

# Entwicklung einer effizienten Entwurfsraumanalyse zur Optimierung der Leistungsaufnahme von Networks-on-Chip

Dissertation

zur Erlangung des akademischen Grades

**Doktor-Ingenieur (Dr.-Ing.)**

der Fakultät für Informatik und Elektrotechnik

der Universität Rostock



vorgelegt von

Gag, Martin, geb. am 16.06.1984 in Güstrow

aus Rostock

Rostock, den 10.6.2014

Gutachter:

- Prof. Dr.-Ing. Dirk Timmermann (Universität Rostock)
- Prof. Dr.-Ing. habil. Helmut Beikirch (Universität Rostock)
- Prof. Dr.-Ing. Frank Sill Torres (Federal University of Minas Gerais, Brasilien)

Tag der Einreichung: 10.6.2014

Tag der Verteidigung: 17.12.2014

# Kurzreferat

Die Fortschritte in der Fertigungstechnik der modernen Mikroelektronik führen zur stetigen Verkleinerung der Transistorstrukturen. Einerseits führt dies zu enormen Leistungssteigerungen und der Miniaturisierung aller datenverarbeitenden Systeme. Andererseits treten neben positiven Effekten auch Probleme aufgrund der immer kleineren Strukturen auf. Die gesteigerte Transistordichte führt unumgänglich zu hohen Verlustleistungsdichten, was durch verschiedene Effekte – vor allem starke Temperaturentwicklung – die Lebensdauer und Zuverlässigkeit der Schaltung beeinträchtigt. Außerdem wird es zum Problem, die große Anzahl zur Verfügung stehender Ressourcen im Entwurfsprozess effizient zu nutzen. Letztendlich müssen Many-Core-Systeme mit flexiblen Schnittstellen der einzelnen Komponenten (IP-Cores) genutzt werden. Die nötige Modularisierung führt zusätzlich zum gesteigerten Kommunikationsbedarf des Systems dazu, dass neue Kommunikationsarchitekturen genutzt werden müssen. Hier stellen Networks-on-Chip (NoC) wegen ihrer guten Skalierbarkeit, Modularität sowie Parallelität eine geeignete Option dar.

Um die identifizierten Probleme zu adressieren, ist es notwendig, während des Entwurfsprozesses eine frühzeitige Auskunft zu den späteren Leistungsdaten des Systems zu erhalten. Dies betrifft im Falle von NoCs vor allem den Datendurchsatz und die Latenz, die die Ausführungszeit der Kommunikationsvorgänge entscheidend bestimmen. Weiterhin ist die Entwicklung des Leistungsumsatzes während des Betriebes interessant, da so Energiebedarf, Temperaturentwicklung und folglich auch Zuverlässigkeit abgeschätzt werden können.

In dieser Arbeit werden diese Aspekte für die Bestandteile eines einfachen NoC untersucht. So wird die Zusammensetzung des Leistungsbedarfs von Links betrachtet, dabei wird besonders auf den Einfluss von Crosstalk-Effekten eingegangen. Weiterhin steht der Router als zentrales NoC-Element im Mittelpunkt der Betrachtungen. Diese Baugruppe wird auf Gatterebene charakterisiert und eine Methodik zur Rückannotation aus höheren Ebenen entwickelt.

---

Die gewonnenen Erkenntnisse werden bei der Entwicklung eines Systemsimulators auf Transaktionsebene für NoCs in SystemC genutzt. Die geschaffene Simulationsumgebung ist um ein Vielfaches schneller als vergleichbar genaue Lösungen auf RT-Ebene und dazu geeignet ein Many-Core-System während des Entwurfs zu charakterisieren sowie dabei die Laufzeit von Anwendungen, Verlustleistungsverteilung und Temperaturverteilung zu ermitteln. Um die zu untersuchenden Anwendungen festzulegen, werden zunächst Task Graphen genutzt, mit denen sich unterschiedlich kommunikationsintensive und parallele Anwendungen nachbilden lassen. Eine Entwurfsraumanalyse zu einigen NoC-Parametern wird beispielhaft durchgeführt und zeigt, dass es optimale Systemgrößen für bestimmte Anwendungen gibt, die nach dem Power-Delay-Produkt bewertet werden. Somit ist eine effiziente Entwurfsraumanalyse auf Systemebene möglich, mit der sich eine optimale Kommunikationsarchitektur für den jeweiligen Anwendungsfall finden lässt. Zusätzliche Möglichkeiten bieten sich durch das integrierte Temperaturmodell, mit dem sich die Ausbreitung der durch Leistungsumsatz entstehenden Wärme ermitteln lässt. Beispielsweise lässt sich ein temperaturabhängiges Systemmanagement bereits während der Systemsimulation im frühen Entwurfsstadium einbinden.

# Abstract

The sustained miniaturization of transistor structures is achieved by progress in manufacturing technology of modern microelectronics. On the one hand, this leads to enormous increases in performance and miniaturization of all data processing systems. On the other hand, problems occur due to decreasing structure sizes. The increased transistor density inevitably leads to high power densities, which affect the life time and reliability of the circuit by various effects – especially high temperatures. In addition, the large number of available resources leads to problems implementing an efficient design process. Hence, many-core systems need to be used with flexible interfaces of the individual IP-cores. The necessary modularization and increased communication needs of huge systems induce the need of new communication architectures. Networks-on-Chip (NoC) provide good scalability, modularity, and high parallel bandwidth. Therefore the utilization of NoCs represent an optimal option to address these problems.

However, it is necessary to obtain information regarding system performance during the early design flow. For NoCs, especially throughput and latency determining the execution time of the communication, are key benchmarks. Furthermore, the development of power dissipation during operation is interesting since it is a prerequisite for the estimation of energy demand, temperature distribution, and consequently reliability.

In this work, the aspects of the components of a simple NoC are examined. Thus, the composition of the power consumption of links and especially the influence of crosstalk effects is considered. Furthermore, the NoC router is a key element of the analysis. The router is extensively characterized at gate level. A methodology for backannotation from higher levels is developed.

The findings are used in developing a system simulator for NoCs at transaction level in SystemC. The created simulation environment outperforms comparable solutions at RT level, is suitable to characterize a many-core system during the design and to gather performance data of applications, power dissipation distribution, and temperature distribution. To provide applications for examination, task graphs are used. Thus, applications with

---

different grades of communication-intensity and parallelism can be simulated. A design space exploration of several NoC parameters is performed as an example and shows that there are optimal system sizes for specific applications, which are valued according to the power delay product. Thus, an efficient design space analysis is possible at system level, with which optimal communication architectures for a given application can be identified. Additional opportunities are available through the integrated temperature model. The propagation of the resulting heat due to power dissipation can be determined. For example, a temperature aware system management may be integrated during simulation in the early design stages.

# Danksagung

An dieser Stelle ergreife ich die Gelegenheit, mich bei meinen Eltern zu bedanken, die mich stets aufopferungsvoll während meines gesamten Bildungswegs begleitet haben. Ohne diese Unterstützung wäre die Verfassung dieser Arbeit nicht möglich gewesen.

Weiterhin danke ich meinem Doktorvater Prof. Dirk Timmermann, der mir durch das entgegengebrachte Vertrauen und die Anstellung am Institut für Angewandte Mikroelektronik und Datentechnik die freie und selbstverantwortliche Arbeit ermöglichte, die zu der vorliegenden Promotionsschrift führte.

Darüber hinaus danke ich Tim Wegner und Philipp Gorski sowie allen Kollegen und Studenten, die an dieser Arbeit beteiligt sind. Schlussendlich richte ich ein großes Dankeschön an alle Mitarbeiter des Instituts für Angewandte Mikroelektronik und Datentechnik, die mir ein Arbeiten in freundlicher und netter Atmosphäre ermöglichten.





# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>xi</b>
<b>Tabellenverzeichnis</b>	<b>xiii</b>
<b>Abkürzungsverzeichnis</b>	<b>xv</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Zielsetzung . . . . .	1
1.2 Struktur der Arbeit . . . . .	3
<b>2 Grundlagen mikroelektronischer Kommunikationssysteme</b>	<b>7</b>
2.1 CMOS-Technologie . . . . .	7
2.2 Leistungsaufnahme und Energieumsatz . . . . .	9
2.2.1 Temperaturabhängigkeit . . . . .	11
2.2.2 Entwicklungen und Tendenzen . . . . .	13
2.2.3 Methoden zur Verringerung von Energie- und Leistungsumsatz . .	14
2.3 Zuverlässigkeit und Robustheit . . . . .	15
2.3.1 Temperaturabhängigkeit von Ausfallursachen . . . . .	19
2.3.2 Methoden zur Steigerung der Robustheit . . . . .	23
2.4 Entwurf . . . . .	25
2.5 Kommunikationsarchitekturen – Networks-on-Chip . . . . .	27
2.5.1 Forschung . . . . .	31
2.5.2 Anwendung . . . . .	33
2.6 Implikationen und Ziele der Arbeit . . . . .	35
<b>3 NoC-Links</b>	<b>37</b>
3.1 Grundlagen . . . . .	37
3.2 Abschätzung von Leistungs- und Flächenbedarf . . . . .	40

3.3	Untersuchung des Crosstalk-Effektes . . . . .	41
3.3.1	Messung mittels digitaler Hardware . . . . .	43
3.3.2	Einfluss auf den Leistungsumsatz . . . . .	51
3.3.3	Folgen und Gegenmaßnahmen . . . . .	57
3.4	Fazit . . . . .	59
<b>4</b>	<b>NoC-Router</b>	<b>61</b>
4.1	Grundlagen . . . . .	62
4.2	Verlustleistungsschätzung . . . . .	65
4.3	Verwendetes Router-Design . . . . .	66
4.3.1	Verbesserungen des Router-Designs . . . . .	68
4.4	Performance . . . . .	69
4.5	Fläche . . . . .	72
4.6	Leistungsaufnahme . . . . .	74
4.6.1	Interpolation der Parameter . . . . .	80
4.6.2	Vergleich der Ergebnisse . . . . .	84
4.7	Fazit . . . . .	87
<b>5</b>	<b>NoC-Modell auf Systemebene</b>	<b>91</b>
5.1	Bestehende NoC-Modelle . . . . .	92
5.2	Implementierung des SystemC-Modells . . . . .	93
5.3	Erzeugung von Verkehrsmustern . . . . .	97
5.4	Task Mapping . . . . .	99
5.5	Integration des Verlustleistungsmodells . . . . .	101
5.6	Integration des Temperaturmodells . . . . .	102
5.7	Anwendung in der Entwurfsraumanalyse . . . . .	106
5.8	Fazit . . . . .	115
<b>6</b>	<b>Zusammenfassung</b>	<b>119</b>
6.1	Ergebnisse der Arbeit . . . . .	119
6.2	Kritik und Ausblick . . . . .	122
	<b>Literaturverzeichnis</b>	<b>125</b>
		<b>137</b>

# Abbildungsverzeichnis

1.1	Kapitelstruktur der Arbeit . . . . .	4
2.1	Temperaturabhängiger Leck- und Kurzschlussstrom . . . . .	12
2.2	Entwicklung der Verlustleistungsdichte und der Transistordichte . . . . .	14
2.3	X-Modell zur Fehlermodellierung . . . . .	18
2.4	Temperatureinfluss auf Defektmechanismen . . . . .	20
2.5	Methoden zur Steigerung der Zuverlässigkeit . . . . .	24
2.6	Hardware-Software-Gap und Design-Productivity-Gap . . . . .	26
2.7	Übersicht verschiedener Kommunikationsarchitekturen . . . . .	29
2.8	Leistungsvergleich verschiedener Kommunikationsarchitekturen . . . . .	30
2.9	Anzahl der Suchergebnisse für NoC in IEEE Xplore . . . . .	32
2.10	Gartner Hype Cycle für Networks-on-Chip . . . . .	32
3.1	Schematische Darstellung einer Leitungsebene innerhalb eines Chips . . . . .	39
3.2	SPICE-Simulation des Crosstalk-Effekts . . . . .	44
3.3	Messschaltung zur Bestimmung der Verzögerung durch Crosstalk-Effekte . . . . .	46
3.4	Auswirkungen der verschiedenen Crosstalk-Muster auf die Oszillator-Frequenzen . . . . .	50
3.5	Schaltwahrscheinlichkeit unterschiedlicher unkomprimierter Daten bei verschiedenen Wortbreiten . . . . .	54
3.6	Durchschnittliche Energie für die Übertragung eines Bits über 200 $\mu\text{m}$ . . . . .	56
4.1	Bestandteile eines Ports des NoC-Routers . . . . .	67
4.2	Performance des Routers mit Zielplattform FPGA und ASIC . . . . .	71
4.3	Flächenbedarf und Frequenz des Routers für FPGA . . . . .	73
4.4	Flächenbedarf der verschiedenen Router-Varianten am Beispiel der 32 nm-Technologie . . . . .	74

4.5	Flächenbedarf der unterschiedlichen ASIC-Technologien . . . . .	75
4.6	Aufschlüsselung der dynamischen Leistungsaufnahme des NoC-Routers	76
4.7	Zusammenhang von Flächenbedarf und dynamischer Leistungsaufnahme des Routers mit und ohne Clock Gating . . . . .	77
4.8	Dynamische Leistungsaufnahme eines Flits – sortiert nach Eingangsport .	79
4.9	Dynamische Leistungsaufnahme eines Flits – sortiert nach Ausgangsport	79
4.10	Anteil des statischen bzw. dynamischen Leistungsumsatzes . . . . .	80
4.11	Statische Leistungsaufnahme der für 100 MHz und 1000 MHz synthetisier- ten Router . . . . .	81
4.12	Abhängigkeit der dynamischen Leistungsaufnahme von der gewählten Betriebsfrequenz . . . . .	82
4.13	Abhängigkeit der dynamischen Leistungsaufnahme von der gewählten Bitbreite der Links . . . . .	83
4.14	Abhängigkeit der dynamischen Leistungsaufnahme von der gewählten Puffertiefe des FIFO-Speichers . . . . .	84
4.15	Vergleich der Leistungsaufnahme für einen NoC-Router aus den Synthese- Ergebnissen und analytischen Abschätzungen von DSENT und McPAT . .	86
4.16	Grundlegende Methodik, die Ergebnisse zum Leistungsumsatz aus der Simulation auf Gatterebene auf Systemebene zu nutzen . . . . .	89
5.1	Grundbestandteile des TLM-Simulators . . . . .	95
5.2	Simulationsgeschwindigkeit des RTL- sowie des TLM-Modells bei verschie- denen NoC-Größen . . . . .	97
5.3	Simulationsgeschwindigkeit des RTL- sowie TLM-Simulators bei verschie- denen Injektionsraten . . . . .	98
5.4	Umsetzung des Temperaturmodells . . . . .	104
5.5	Ausführungszeit einer Anwendung auf unterschiedlich großen Systemen	108
5.6	Energiebedarf verschiedener NoC-Größen . . . . .	109
5.7	EDP verschiedener NoC-Größen . . . . .	110
5.8	EDP für Applikation 1 . . . . .	112
5.9	EDP für Applikation 2 . . . . .	114
5.10	EDP für Applikation 3 . . . . .	115
5.11	Temperaturverlauf . . . . .	116

# Tabellenverzeichnis

2.1	Methoden zur Leistungsreduktion . . . . .	16
3.1	Mögliche MCF einer Leitung beim Schalten von 0 auf 1 . . . . .	42
3.2	Crosstalk-Muster für das Schalten der Opferleitung von 0 auf 1 . . . . .	43
3.3	Frequenzen der unterschiedlichen Crosstalk-Muster . . . . .	49
4.1	Entscheidungstabelle bei XY-Routing . . . . .	63
4.2	Übersicht der wählbaren Designparameter . . . . .	68
4.3	Verwendete Implementierungen des NoC-Routers . . . . .	70
4.4	Vergleich von DSENT, McPAT und dem Ansatz dieser Arbeit . . . . .	87
5.1	Dualität zwischen thermischem und elektrischem Modell . . . . .	103
5.2	Verwendete Task-Graph Sets . . . . .	111
5.3	Effizienz der Anwendungen unter Verwendung der verschiedenen Systemparameter . . . . .	113



# Abkürzungsverzeichnis

AMS .....	<u>A</u> nalog <u>M</u> ixed <u>S</u> ystem
ARM .....	<u>A</u> dvanced <u>R</u> ISC <u>M</u> achines
ASIC .....	<u>A</u> pplication <u>S</u> pecific <u>I</u> C
BEAM .....	<u>B</u> order <u>E</u> nhanced <u>M</u> esh
CAC .....	<u>C</u> rosstalk <u>A</u> voidance <u>C</u> ode(s)
CLB .....	<u>C</u> onfigurable <u>L</u> ogic <u>B</u> lock
CMOS .....	<u>C</u> omplementary <u>M</u> etal <u>O</u> xide <u>S</u> emiconductor
CPU .....	<u>C</u> entral <u>P</u> rocessing <u>U</u> nit
DPG .....	<u>D</u> esign- <u>P</u> roductivity- <u>G</u> ap
DSENT .....	<u>D</u> esign <u>S</u> pace <u>E</u> xploration for <u>N</u> etwork <u>T</u> ool
DSP .....	<u>D</u> igital <u>S</u> ignal <u>P</u> rocessor
DVFS .....	<u>D</u> ynamic <u>V</u> oltage <u>F</u> requency <u>S</u> caling
EDA .....	<u>E</u> lectronic <u>D</u> esign <u>A</u> utomation
EDP .....	<u>E</u> nergy- <u>D</u> elay- <u>P</u> rodukt
ELN .....	<u>E</u> lectric <u>L</u> inear <u>N</u> etwork
EM .....	<u>E</u> lektromigration
EMI .....	<u>E</u> lectromagnetic <u>I</u> nterference
FEM .....	<u>F</u> inite- <u>E</u> lemente- <u>M</u> ethode
FIFO .....	<u>F</u> irst- <u>I</u> n- <u>F</u> irst- <u>O</u> ut
Flit .....	<u>F</u> low <u>C</u> ontrol <u>D</u> igit
FPGA .....	<u>F</u> ield <u>P</u> rogrammable <u>G</u> ate <u>A</u> rray
GPP .....	<u>G</u> eneral <u>P</u> urpose <u>P</u> rocessor
HCI .....	<u>H</u> ot <u>C</u> arrier <u>I</u> njection
HPC .....	<u>H</u> igh <u>P</u> erformance <u>C</u> omputing
HSG .....	<u>H</u> ardware- <u>S</u> oftware- <u>G</u> ap
IP .....	<u>I</u> ntellectual <u>P</u> roperty
ITRS .....	<u>I</u> nternational <u>T</u> echnology <u>R</u> oadmap for <u>S</u> emiconductors

JEDEC .....	<u>J</u> oint <u>E</u> lectro <u>n</u> <u>D</u> evice <u>E</u> ngineering <u>C</u> ouncil
JPEG .....	<u>J</u> oint <u>P</u> hotographic <u>E</u> xperts <u>G</u> roup
LUT .....	<u>L</u> ook <u>U</u> p <u>T</u> able
MCF .....	<u>M</u> iller <u>C</u> oupling <u>F</u> aktor
MC .....	<u>M</u> emory <u>C</u> ontroller
McPAT .....	<u>M</u> ulticore <u>P</u> ower, <u>A</u> rea, and <u>T</u> iming
NBTI .....	<u>N</u> egative <u>B</u> ias <u>T</u> emperature <u>I</u> nstability
NI .....	<u>N</u> etwork <u>I</u> nterface
NoC(s) .....	<u>N</u> etwork(s)-on- <u>C</u> hip
OMAP .....	<u>O</u> pen <u>M</u> ultimedia <u>A</u> pplication <u>P</u> lattform
PBTI .....	<u>P</u> ositive <u>B</u> ias <u>T</u> emperature <u>I</u> nstability
PCM .....	<u>P</u> uls- <u>C</u> ode- <u>M</u> odulation
PDP .....	<u>P</u> ower- <u>D</u> elay- <u>P</u> rodukt
PLB .....	<u>P</u> rocessor <u>L</u> ocal <u>B</u> us
PTM .....	<u>P</u> redictive <u>T</u> echnology <u>M</u> odel
QoS .....	<u>Q</u> uality of <u>S</u> ervice
QPI .....	<u>Q</u> uick <u>P</u> ath <u>I</u> nterconnect
RAM .....	<u>R</u> andom <u>A</u> ccess <u>M</u> emory
Req/Ack .....	<u>R</u> equ <u>e</u> st- <u>A</u> cknowledgment- <u>P</u> rotokoll
RGT-Regel .....	<u>R</u> eaktionsgeschwindigkeit- <u>T</u> emperatur- <u>R</u> egel
RISC .....	<u>R</u> educed <u>I</u> nstruction <u>S</u> et <u>C</u> omputing
RTL .....	<u>R</u> egister <u>T</u> ransfer <u>L</u> evel
SCC .....	<u>S</u> ingle- <u>C</u> hip <u>C</u> loud <u>C</u> omputer
SoC(s) .....	<u>S</u> ystem(s)-on- <u>C</u> hip
SPICE .....	<u>S</u> imulation <u>P</u> rogram with <u>I</u> ntegrated <u>C</u> ircuit <u>E</u> mphasis
TAO .....	<u>T</u> ransaction <u>O</u> bject
TDDb .....	<u>T</u> ime <u>D</u> ependent <u>D</u> ielectric <u>B</u> reakdown
TGFF .....	<u>T</u> ask <u>G</u> raphs <u>F</u> or <u>F</u> ree
TGS .....	<u>T</u> ask <u>G</u> raph <u>S</u> et
TLM .....	<u>T</u> ransaction <u>L</u> evel <u>M</u> odelling
UCF .....	<u>U</u> ser <u>C</u> onstraint <u>F</u> ile
VC .....	<u>V</u> irtual <u>C</u> hannel
VHDL .....	<u>V</u> ery <u>H</u> igh <u>S</u> peed <u>I</u> ntegrated <u>C</u> ircuit <u>H</u> ardware <u>D</u> escription <u>L</u> anguage



# Kapitel 1

## Einleitung

### 1.1 Motivation und Zielsetzung

Die Verbreitung von mikroelektronischen Komponenten stieg in der jüngsten Vergangenheit rasant an. Leistungsfähige Mikrochips finden sich nicht nur in Rechenzentren, sondern auch in Haushaltsgeräten sowie mobilen Unterhaltungs- und Kommunikationsgeräten. In der Folge steckt in praktisch jedem Produkt Elektronik, sei es in Form von passiven Funkchips zur Identifikation während Lagerung, Transport und Verkauf. Vor allem durch die Verbreitung von mobilen Kommunikationsgeräten – momentan gibt es ungefähr so viele Mobiltelefone wie Menschen auf der Welt [ITU13] – stieg die Durchsetzung des Alltags mit elektronischen Geräten in den letzten 20 Jahren enorm an.

Um diese starke Verbreitung zu erreichen, mussten mikroelektronische Produkte universell einsetzbar, leistungsfähig und kostengünstig werden. Dies wurde durch Verbesserungen in den Fertigungsverfahren erreicht, die eine Entwicklung der Strukturgrößen nach dem Mooreschen Gesetz [Moo65] ermöglichten. So war es möglich, die Komplexität der Schaltungen etwa alle 18 Monate zu verdoppeln. Dies gestattete die Steigerung der Leistungsfähigkeit und Effizienz von Prozessoren durch z. B. höhere Betriebsfrequenzen, größere Befehlssätze und tiefere Pipelines.

Die verwendeten Architekturen wandelten sich von diskreten Systemen, die aus mehreren Chips bestanden, durch die Integration von immer mehr Systemfunktionen zu so genannten Systems-on-Chip (SoC). In ein SoC werden mittlerweile nicht mehr nur ein Prozessor sondern mehrere und verschiedene Recheneinheiten integriert. Diese Entwick-

lung über Multi- und hin zu Many-Core-Architekturen stellt besondere Anforderungen an die Kommunikationssysteme innerhalb des Chips. Neben hohen Bandbreiten und niedrigen Latenzen stellt das Erreichen einer hohen Flexibilität und Produktivität im Entwurfsprozess eine Herausforderung dar. Größere homogene Many-Core-Chips machen genauso wie heterogene SoCs den Einsatz von paketvermittelnden on-Chip-Netzwerken attraktiv. Diese so genannten Networks-on-Chip (NoC) stellen eine hohe Gesamtbandbreite zur Verfügung und bieten eine hohe Flexibilität, Effizienz sowie Skalierbarkeit und Wiederverwendbarkeit während des Systementwurfs.

Die Entwicklung der Integrationsdichten nach dem Moorschen Gesetz hat jedoch auch Nachteile. Das Hauptproblem bilden die enormen Verlustleistungsdichten, die durch immer mehr Schaltungselemente auf dem gleichen Raum entstehen. Dies hat dazu geführt, dass Performancesteigerungen nicht mehr durch das Erhöhen der Taktraten erreicht werden können. Stattdessen wird auf Parallelisierung und Hardware-Beschleunigung gesetzt. Aber selbst wenn das System aus vielen Prozessorkernen mit moderaten Taktfrequenzen besteht, können inzwischen nicht mehr alle Teile des Systems gleichzeitig voll ausgelastet werden, da die gesamte Verlustleistung für das Kühlsystem zu hoch wäre. Die Folge ist so genanntes Dark Silicon [EBSA<sup>+</sup>11] – also das zwangsweise Abschalten von einigen Systemteilen bei zu hoher Auslastung.

Eine hohe Verlustleistung führt zwangsläufig zu starker Wärmeentwicklung. Dies beschleunigt vor allem die chemischen Prozesse innerhalb des Chips, die als Alterung bezeichnet werden und die Leistungsfähigkeit sowie letztendlich die Funktionalität der Schaltung beeinträchtigen. Auch die ungleichmäßige Erwärmung durch inhomogene Aktivität im System kann durch die Entwicklung von Hotspots und daraus entstehenden mechanischen Spannungen die Zuverlässigkeit eines Chips beeinträchtigen. Außerdem führt eine insgesamt hohe maximale Leistungsaufnahme dazu, dass das Kühlsystem für den Chip entsprechend groß ausgelegt werden muss. Dies und ein eventuell hoher Energieverbrauch führen zu Problemen, solche Produkte in mobilen Geräten zu betreiben. Kleine Bauformen und lange Akkulaufzeiten sind hier essentielle Anforderungen.

Mikroelektronische Systeme zu entwerfen ist aufgrund der immensen zur Verfügung stehenden Transistorzahl je Chip sehr aufwändig. Trotzdem müssen die Entwicklungszeiten gering gehalten werden, um konkurrenzfähige Produkte anbieten zu können. In der Folge ist ein Entwurf des Gesamtsystems in möglichst kurzer Zeit nötig. Dies bedeutet, dass große Teile der Software bereits erstellt werden müssen, während die Hardware

sich noch im Entwurfs- und Fertigungsprozess befindet. Ein wichtiges Erfordernis für dieses Vorgehen ist, Änderungen an der Systemarchitektur möglichst früh im Entwurfsstadium einzuführen, damit wenig andere Entwicklungsschritte beeinflusst werden. Dazu müssen möglichst genaue Abschätzungen z. B. zur Leistungsaufnahme der einzelnen Komponenten bereits auf Systemebene erfolgen.

Aus diesen Gründen soll in der vorliegenden Arbeit die Leistungsaufnahme von NoCs auf unterschiedlichen Ebenen untersucht werden. Dabei wird darauf Wert gelegt, eine möglichst genaue Abschätzung schon auf hoher Entwurfsebene vorzunehmen und dabei eine effiziente Simulationsgeschwindigkeit des Systems beizubehalten. Auf diese Weise wird eine Möglichkeit geschaffen, die Nutzung von NoCs und deren Leistungsaufnahme während einer Entwurfsraumanalyse von Many-Core-Systemen mit einzubeziehen. Weiterhin werden die möglichen Folgen von zu hoher und ungünstig verteilter Leistungsaufnahme in Form von Temperaturentwicklung sowie Einschränkungen der Zuverlässigkeit NoC-basierter Systeme untersucht.

## 1.2 Struktur der Arbeit

Diese Dissertationsschrift unterteilt sich in 6 Kapitel. Die Struktur ist in Abbildung 1.1 dargestellt. Dabei sind Kapitel, die grundlegende bekannte Aspekte behandeln, hellgrün markiert. Abschnitte die hauptsächlich auf eigenen Untersuchungen beruhen, sind in dunkelgrün dargestellt.

Im ersten Kapitel wird ein grober Überblick über die behandelte Thematik und eine Hinführung zu den bearbeiteten Gegenständen dieser Arbeit gegeben. Außerdem ist die in diesem Abschnitt erklärte Struktur der Arbeit Thema.

Um die angewandten Methoden zu verstehen und einen erweiterten Einblick in die Hintergründe der Problematiken zu gewähren, werden im zweiten Kapitel in mehreren Abschnitten die Grundlagen von mikroelektronischen Systemen beleuchtet. Neben dem speziellen Eingehen auf Leistungsumsatz und Zuverlässigkeit solcher Systeme ist auch ein Abschnitt zu Kommunikationsarchitekturen enthalten, in dem die Verwendung von NoCs untersucht wird.

Die einzelnen Bestandteile eines NoC werden in den folgenden Kapiteln 3 und 4 genauer untersucht. Dabei wird zunächst auf die Links eingegangen, ihr Einfluss auf die Leistungsaufnahme des NoC hergeleitet und das spezielle Phänomen des Crosstalks

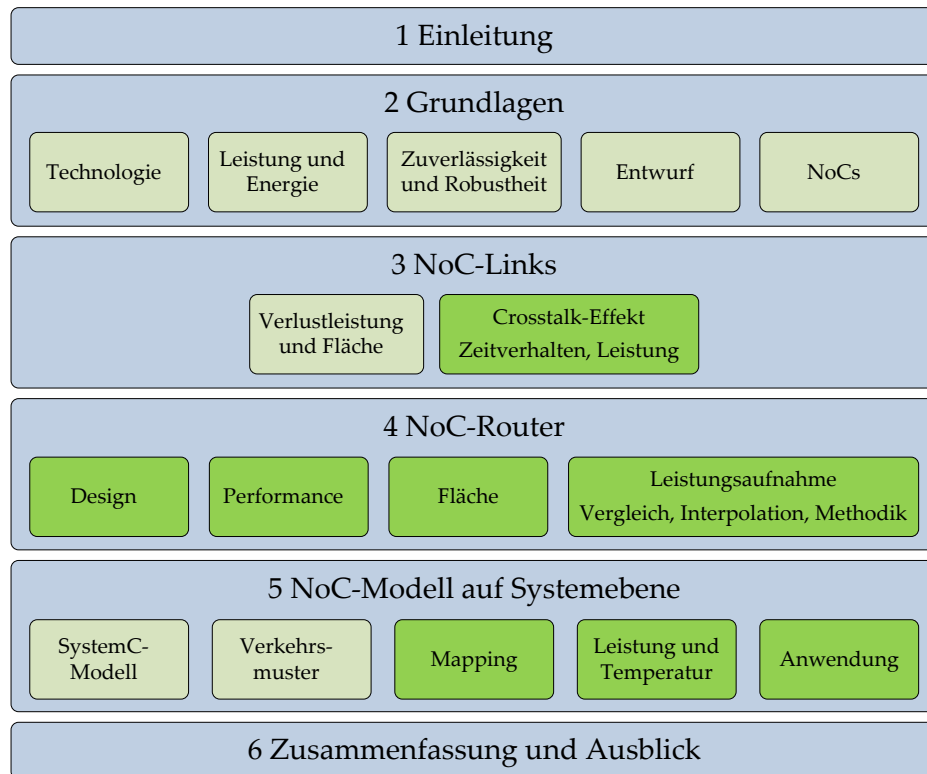


Abbildung 1.1: Kapitelstruktur der Arbeit – Grundlagen und auf anderen Arbeiten basierende Teile: hellgrün, eigene Untersuchungen: dunkelgrün

näher betrachtet. Im vierten Kapitel stehen die NoC-Router im Mittelpunkt. Hier wird eine eigens entworfene Router-Architektur, deren Leistungsfähigkeit durch Parameter einstellbar ist, vor allem hinsichtlich der Leistungsaufnahme untersucht.

Im fünften Kapitel wird eine Simulationsumgebung für NoC-basierte Systeme auf Systemebene vorgestellt. Dabei wird auf die effiziente und dennoch exakte Implementierung genauso eingegangen wie auf den Betrieb des Systems mit verschiedenen Eingangsdaten und die Verteilung der Softwareprozesse innerhalb des Many-Core-Systems. In weiteren Abschnitten geht es um die Erweiterung um Modelle der Leistungsaufnahme und der Temperaturentwicklung. Letztendlich wird an einigen Beispielen gezeigt, wie eine Entwurfsraumanalyse mit dem Schwerpunkt der Leistungsaufnahme ermöglicht wird. Anschließend folgt die Demonstration einer Anwendung des implementierten Verlustleistungsmodells durch Darstellung der resultierenden Temperaturverteilung.

Im letzten Kapitel wird die Arbeit zusammengefasst und denkbare Anwendungsgebiete werden identifiziert. Möglichen Schwächen werden Verbesserungs- und Erweiterungsmöglichkeiten entgegengehalten.



# Kapitel 2

## Grundlagen mikroelektronischer Kommunikationssysteme

### 2.1 CMOS-Technologie

Auf der Complementary Metal Oxide Semiconductor (CMOS)-Technologie basieren heutzutage die meistgenutzten Halbleiterprozesse sowie die verbreitetste Logikfamilie. Die prinzipielle Entwicklung von CMOS-Technologien ist bisher vor allem durch drei Trends gekennzeichnet – Strukturverkleinerung, Vergrößerung der Chipfläche und komplexere Herstellungsprozesse. Die bedeutendste Veränderung ist dabei die Verkleinerung der Strukturen. In deren Folge steigen die Schaltgeschwindigkeiten der Schaltungen durch kleinere zu ladende Kapazitäten. Außerdem können immer mehr Komponenten auf kleinerem Raum integriert werden. Dies verkürzt die Signalwege zwischen den Baugruppen. Insgesamt werden integrierte Schaltungen immer leistungsfähiger.

Während die Integrationsdichte exponentiell anwächst, werden außerdem die Chipgrößen leicht erhöht. Nachdem in den Anfängen von CMOS nur einige Transistoren auf einem Chip untergebracht wurden, können durch diese beiden Entwicklungen momentan einige Milliarden Transistoren auf einmal integriert werden. Den exponentiellen Anstieg der zur Verfügung stehenden Ressourcen hat Gordon Moore bereits 1965 als Regelmäßigkeit erkannt [Moo65]. Die Verdopplung der Transistorzahl wurde zunächst alle 18 Monate als Ziel der technologischen Weiterentwicklungen verfolgt.

Das Erreichen dieses Ziels stellt jedoch immer wieder ein Problem dar, so dass sich die angedachte Verdoppelung momentan nur noch ca. alle zwei bis drei Jahre realisieren lässt [ITR12]. Diesen Fortschritten liegt eine ständige Weiterentwicklung der Fertigungstechnologien und der verwendeten Materialien zugrunde. Beispielsweise muss die verwendete Wellenlänge während der Belichtungsprozesse an feinere Maskenstrukturen angepasst werden. Die filigranere Fertigung erfordert Materialien mit geeigneten Eigenschaften, beispielsweise Metalle mit höherem Leitwert und Dielektrika mit geringerer Permittivität. Weiterhin sind Entwicklungen im Bereich der Strukturen nötig, die z. B. zu dreidimensionalen Transistorformen führen. Die komplexer werdenden Herstellungsprozesse führen zu einer größeren Fehlerrate und zu stark variierenden Parametern der Schaltungen. Bessere Test-Verfahren und so genanntes Binning [SLSN98] – das Sortieren von unterschiedlich funktionalen Chips zu unterschiedlichen Produkten – wurden nötig um die Ausbeute auf ökonomisch sinnvollem Niveau zu halten.

Mit den genannten drei Entwicklungen gehen allerdings auch Probleme einher, die auf höheren Entwurfsebenen bearbeitet werden müssen. Die kleineren Dimensionen sorgen vor allem für größere Stromdichten und höhere Feldstärken, die wiederum zu größeren Leckströmen der Bauteile führen. Dies in Verbindung mit der Vergrößerung der Chipflächen verursacht eine höhere Leistungsaufnahme, die in einer gesteigerten Hitzeentwicklung und größeren Temperaturunterschieden innerhalb des Chips mündet. Es entsteht eine größere mechanische Belastung der Materialien und Ausfallprozesse werden beschleunigt. Folglich kann es schneller zu einem Defekt kommen. Dies ist eine Auswirkung der technologischen Verbesserungen, die sich auf das Gesamtsystem des Mikrochips bezieht und somit bereits während des Systementwurfs berücksichtigt werden muss.

Die Strukturverkleinerung ist in erster Linie für die Steigerung der Anzahl der integrierbaren Transistoren nützlich. Leitungen in schrumpfenden CMOS-Prozessen besitzen hingegen einen tendenziell höheren Widerstand, wenn der Leitungsquerschnitt sinkt. Das bedeutet, die Leitungsverzögerung steigt und fällt bei der erzielbaren Performance mehr ins Gewicht. Deshalb werden die Leitungen im Allgemeinen nicht in gleicher Weise skaliert. Da aber die Transistordichte steigt, werden mehr Leitungen auf dem gleichen Raum benötigt. Diese werden zwar im Durchschnitt etwas kürzer, globale Leitungen sind davon jedoch nicht betroffen. Sie dienen dazu, einzelne Teilmodule des Systems zu verbinden und hängen nicht direkt von der Strukturgröße, sondern eher von der Position



der Komponenten zueinander ab. Die steigenden Leitungsverzögerungen erfordern das Einfügen von mehr Treiberschaltungen, um die Anforderungen an die Signalintegrität, also Zeitanforderungen, Pegelstabilität sowie Flankensteilheit erfüllen zu können. Dies erhöht die Leistungsaufnahme. Da die Leitungen auch dichter nebeneinander platziert werden müssen, werden die Kapazitäten insbesondere zwischen den Leitungen größer. Um dieses Problem abzumildern werden neue Materialien als Dielektrikum zwischen den Leitungen genutzt. Dies geht soweit, dass zukünftig Luftspalte zwischen den leitenden Metallen realisiert werden [Par11].

Zusammenfassend lässt sich feststellen, dass die Entwicklungen im technologischen Bereich neben Performancesteigerungen vor allem die Probleme der hohen Leistungsaufnahme und der Zuverlässigkeit verschärft haben. Für zukünftige Entwicklungen wird es sehr wichtig sein, diese Aspekte besonders in höheren Entwurfsebenen zu beachten.

## 2.2 Leistungsaufnahme und Energieumsatz

Die Leistungsaufnahme gibt an, wie viel Energie in einer bestimmten Zeit von einem Verbraucher umgesetzt wird. Da nach dem Energieerhaltungssatz keine Energie verloren geht, wird diese dabei von elektrischer in thermische oder chemische Energie umgewandelt. Die Leistungsaufnahme einer CMOS-Schaltung ist auf zwei grundlegende Mechanismen zurückzuführen. Zum einen müssen die Gate-Kapazitäten über den p-Transistor aufgeladen werden. Die dabei gespeicherte Ladung wird beim Umschalten des Gatterausgangs über den n-Transistor an die Masseleitung entladen. Zum anderen ist die Versorgungsspannung nicht vollständig von der Masse getrennt, dadurch entstehen Leckströme und so genannte Kurzschlussströme. Aus dem Produkt des Gesamtstroms zwischen den Versorgungsleitungen mit der Spannungsdifferenz ergibt sich die Leistungsaufnahme der Schaltung (siehe Gleichung 2.1).

$$P = I_{ges} \cdot V_{DD} = P_{dyn} + P_{sc} + P_{stat} \quad (2.1)$$

$$P_{dyn} = \frac{1}{2} \cdot f \cdot V_{DD}^2 \cdot C_L \quad (2.2)$$

$$P_{sc} = f \cdot t_{sc} \cdot I_{peak} \cdot V_{DD} \quad (2.3)$$

$$P_{stat} = I_{leak} \cdot V_{DD} \quad (2.4)$$

Man unterscheidet die dynamische von der statischen Leistungsaufnahme. Außerdem ist die durch die temporären Kurzschlussströme verursachte Leistungsaufnahme während der Umschaltvorgänge zu erwähnen, die als dynamisches Phänomen betrachtet werden sollte. Dynamische Leistungsaufnahme ( $P_{dyn}$ ) wird durch Zustandswechsel – also durch die Änderung von Signalen – hervorgerufen. Der Vorgang umfasst das Auf- bzw. Entladen der Lastkapazität  $C_L$ . Diese wird von den Gate-Kapazitäten der folgenden Transistoren und von verschiedenen parasitären Kapazitäten (Transistoren, Leitungen) gebildet. Nur bei der Hälfte aller Ausgangsänderungen – also Ent- oder Ladevorgang – wird diese Leistung umgesetzt.

Der direkte Stromfluss zwischen den Versorgungsleitungen verursacht die Leistungsaufnahme durch einen temporären Kurzschluss ( $P_{sc}$ ), wenn p- und n- Transistoren umschalten und so kurze Zeit gleichzeitig leiten. Dies kann jedoch theoretisch in jedem Takt der Fall sein – also doppelt so oft wie bei  $P_{dyn}$ .

Die statische Leistungsaufnahme ( $P_{stat}$ ) wird durch Leckströme verschiedenen Ursprungs hervorgerufen [NC06]. Die wichtigsten sind im Folgenden aufgeführt.

- Subthreshold Leakage ( $I_{sub}$ ): Leckstrom zwischen Drain und Source trotz geschlossenem Kanal, Abhängigkeit von Oxiddicke und Kanallänge, stark temperaturabhängig (Gleichung 2.5)
- Gate Oxide Tunnelling Leakage ( $I_G$ ): vom Gate zum Substrat, Source und Drain; bei offenem und geschlossenem Kanal; nicht temperaturabhängig; steigt stark mit Verringerung der Oxiddicke (Gleichung 2.6)
- Reverse Bias Junction Leakages ( $I_{pn}$ ): von Source und Drain in das Substrat; bei offenem und geschlossenem Kanal; in Sperrrichtung vorgespannter pn- Übergang (Gleichung 2.7)
- Gate Induced Drain Leakage ( $I_{GIDL}$ ): Entstehen von Ladungsträgern durch starke Felder im Bereich von Drain direkt unter dem Gate; bei geschlossenem Kanal; nicht temperaturabhängig
- Gate Current Due To Hot-Carrier Injection: vom Gate zum Substrat; bei geschlossenem Kanal und Schaltübergängen
- Channel Punch Through Current: bei geschlossenem Kanal; zwischen Drain und Source

Den größten Anteil hat in der Regel Subthreshold Leakage [WH10]. Dieser Leckstrom hängt unter anderem von der Größe des Transistors und der Schichtdicke des Gates ab (siehe Gleichung 2.5 und 2.8). Zur Verminderung dieses Effektes werden verschiedene Techniken (siehe Abschnitt 2.2.3) eingesetzt, die z. B. durch das Bereitstellen von Transistoren mit verschiedenen Schwellspannungen unnötig dünne Dielektrika vermeiden oder die Transistoren innerhalb des Gatters besonders geschickt anordnen (Stack-Effekt [NC06]). Ein weiterer großer Einflussfaktor ist die Temperatur.  $F_1$  bis  $F_5$  sind konstante Hilfsfaktoren. Parameter mit hauptsächlichem, da exponentiellem Einfluss sind Schwellspannung ( $V_{th}$ ), Oxiddicke ( $T_{ox}$ ) und Temperatur ( $T$ ).

$$I_{sub} = F_1 \cdot \frac{W}{L} \cdot C_{ox} T^2 \cdot e^{\frac{-qV_{th0}}{nk} (b + \frac{1}{T} (1 + bT_0))} \quad (2.5)$$

$$I_G = F_2 \cdot W \cdot L \cdot \left( \frac{V_{gs}}{T_{ox}} \right)^2 \cdot e^{\frac{-F_4 \cdot T_{ox}}{V_{gs}}} \quad (2.6)$$

$$I_{pn} = I_{diode} \cdot \left[ e^{\frac{F_5 \cdot V_{pn}}{T}} - 1 \right] \quad (2.7)$$

### 2.2.1 Temperaturabhängigkeit

Der dynamische Leistungsumsatz hängt nicht von der Temperatur ab, da die Kapazitäten, die auf- und entladen werden, sowie die Betriebsspannung und Häufigkeit der Ausgangsänderung konstant bleiben. Die Spitzenleistung während des Umschaltvorgangs kann eine gewisse Temperaturabhängigkeit aufweisen, da sich die Leitwerte der Transistoren ändern und der Umschaltvorgang so länger dauert, wenn die Temperatur steigt. Dieser Effekt ist aber vernachlässigbar gering.

Subthreshold-Leakage ist der einflussreichste Leckstrom. Er hängt exponentiell von der Temperatur ab (siehe Abbildung 2.1). Dies führt dazu, dass in Systemen, in denen der Leckstrom die Leistungsaufnahme dominiert, eine Temperatursteigerung unmittelbar zu Problemen durch erhöhte Leistungsaufnahme führen kann.

Der Kurzschlussstrom ist ebenfalls von der Temperatur abhängig. Durch Änderung von  $V_{th}$  und  $\mu$  (siehe Gleichung 2.9 und 2.11) verändert sich auch die Steilheit der Flanken und damit die Zeit in der der Kurzschlussstrom fließen kann. Eine analytische Beschreibung dieses Vorgangs bei modernen Prozesstechnologien ist nicht ohne weiteres möglich. Aus diesem Grund werden häufig empirische Modelle bzw. darauf basierende SPICE-Simulationen genutzt. Diese zeigen eine leichte Temperaturabhängigkeit des Kurz-

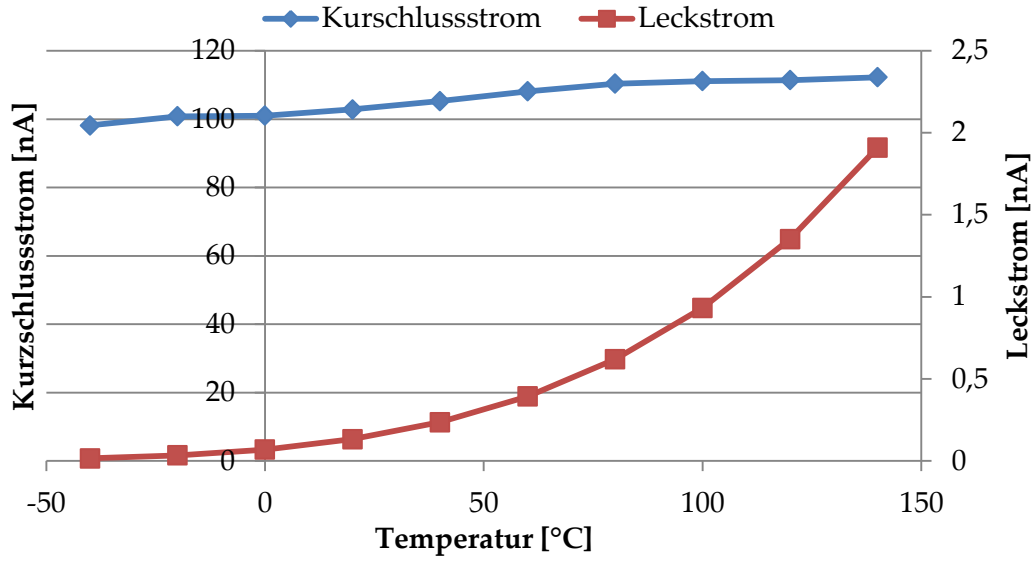


Abbildung 2.1: Temperaturabhängiger Leck- und Kurzschlussstrom eines CMOS-Inverters, SPICE-Simulation mit BSIM4-Transistormodellen bei 45 nm

schlussstroms. Im typischen Temperaturbereich (0-85 °C, siehe Abschnitt 2.3.1) lässt sich demnach ein linearer Zusammenhang vermuten (siehe Abbildung 2.1). Der Einfluss ist insgesamt gering und nicht exponentiell, wie beim Leckstrom.

$$I_{sub} = I_0 \cdot e^{\frac{V_{GS}-V_{th}}{n \cdot V_T}} \left[ 1 - e^{\frac{-V_{DS}}{V_T}} \right] \quad (2.8)$$

$$V_{th} = V_{th0}(1 + b(T - T_0)) \text{ mit } b = -2 \dots -4 \left[ \frac{mV}{K} \right] \quad (2.9)$$

$$V_T = \frac{k \cdot T}{q} \quad (2.10)$$

$$\mu = \mu_0 \cdot \left( \frac{T}{T_0} \right)^a \text{ mit } a = -1,5 \dots -2,5 \quad (2.11)$$

$$I_0 = \frac{W \mu_0 C_{ox} V_T^2 e^{1.8}}{L} \quad (2.12)$$

### 2.2.2 Entwicklungen und Tendenzen

Bei der Betrachtung der Auswirkungen von erhöhtem Leistungs- und Energiebedarf sind ökonomische, ökologische, technische und funktionale Aspekte zu beachten. Aus wirtschaftlicher Sicht entstehen durch hohen Energiebedarf direkte Versorgungskosten, während hoher Leistungsumsatz eine große Hitzeentwicklung zur Folge hat, die unter Umständen durch aktive Kühlösungen abgeführt werden muss. Dies kann wiederum Energiekosten verursachen. Ähnliche Abwägungen liegen der Energieeinsparung aus ökologischer Sicht (Green IT) zugrunde.

Durch die hohen Integrationsdichten sind die Leistungsdichten ebenfalls stark angestiegen und es werden technische Maßnahmen erforderlich, um die Zuverlässigkeit trotz hoher Ströme und Temperaturen zu sichern. Diese Maßnahmen verringern die Performance des Systems oder erhöhen dessen Ressourcenbedarf. Durch zusätzlich nötige Kühlmechanismen steigt außerdem das Gewicht des Produktes, eventuell erhöht sich auch die Geräuschemission. Diese Faktoren grenzen die Einsatzmöglichkeiten vor allem im mobilen Bereich ein. Hier wird auch eine beschränkte Laufzeit durch hohen Energieumsatz zum wichtigen Aspekt. Die angeführten Gründe sorgen dafür, dass ein möglichst geringer Energiebedarf zur Systemanforderung wird. Weitere kritische Punkte sind der Leistungsumsatz sowie die folgenden hohen Leistungsdichten, Temperaturen und resultierende Zuverlässigkeitsprobleme.

Aufgrund dieser Aspekte ist die Höhe des möglichen Leistungsumsatzes begrenzt. Die bisherige Entwicklung lässt sich vor allem anhand des Vergleichs von Leistungs- und Integrationsdichte zeigen (siehe Abbildung 2.2). Während die Transistordichte nach dem Mooreschen Gesetz exponentiell ansteigt, verhält es sich mit der Verlustleistungsdichte analog, bzw. steigt diese zum Teil sogar stärker an, da gleichzeitig die Taktfrequenzen gesteigert werden können. Es wird jedoch absehbar, dass die technische Machbarkeit dieser Skalierung nicht mehr zu gewährleisten ist. Das betrifft das effiziente Abführen der Verlustleistung, um die maximalen Temperaturen im Chip im akzeptablen Rahmen zu halten. Ungefähr ab der 90 nm-Generation (Intel Pentium 4) wird deshalb vermehrt auf Maßnahmen gesetzt, die Leistungsaufnahme zu senken. In der Folge kann bei jüngeren Technologiegenerationen ein leichter Rückgang der Leistungsdichte festgestellt werden.

Es wird auf allen Ebenen daran gearbeitet, den Leistungsumsatz zu verringern (siehe Abschnitt 2.2.3). Die Betriebsfrequenzen lassen sich nicht mehr signifikant steigern, da die nötige, höhere Versorgungsspannung zu einem zu hohen Leistungsanstieg führen würde.

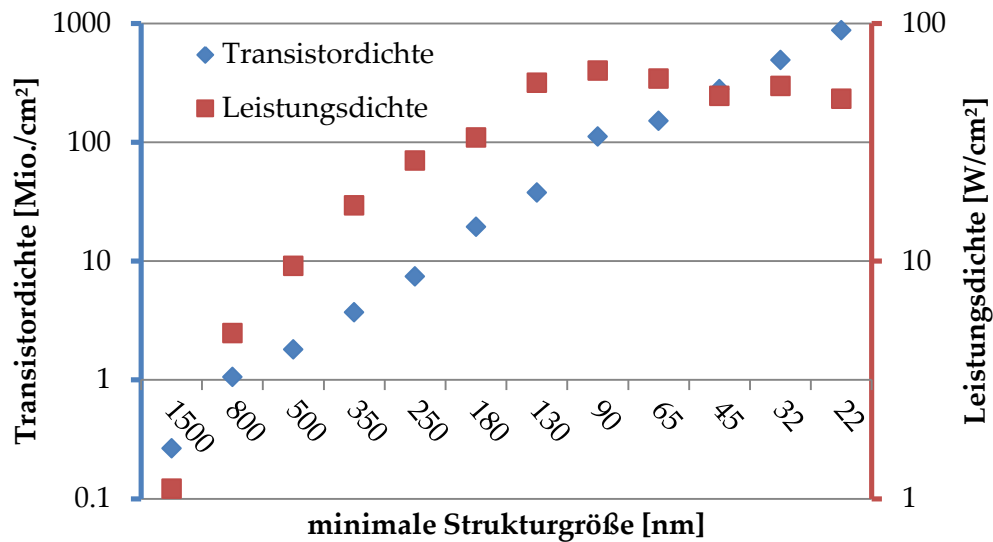


Abbildung 2.2: Entwicklung der Verlustleistungsdichte und der Transistordichte über verschiedene Fertigungstechnologien am Beispiel typischer Mikroprozessoren des Herstellers Intel [CPU13]

Stattdessen wird auf Parallelisierung, große schnelle Speicher auf dem Chip (Cache) und effizientere Mikroarchitekturen gesetzt.

### 2.2.3 Methoden zur Verringerung von Energie- und Leistungsumsatz

Um Energie einzusparen und die Leistungsaufnahme zu reduzieren, sind vor allem ab der 90 nm-Technologie (siehe Abschnitt 2.2.2) viele verschiedene Techniken genutzt worden. Auch für die Zukunft werden in der International Roadmap for Semiconductors (ITRS) weitere Methoden vorgeschlagen. In Tabelle 2.1 ist außerdem angegeben, wie groß die Einsparung im Bereich dynamischer respektive statischer Leistungsaufnahme eingeschätzt wird. Die Werte sind als Verhältnis des bisherigen zum neuen Leistungsbedarf zu interpretieren. Eine Einsparung entspricht also Werten über eins und ein höherer Leistungsbedarfs entspricht Werten unter eins.

Insgesamt werden verschiedene Ansätze auf allen Ebenen des Entwurfs und des Systemmanagements in Betracht gezogen. Während z. B. statisches und adaptives Body Biasing – also das Ändern der Substratspannung einiger Transistoren oder Transistorgruppen –

Techniken auf technologischer Ebene sind, stellt das Anbieten von Low Power Bibliotheken eine Möglichkeit auf Zellebene dar. Beim Power Gating, wenn ganze Gattergruppen von der Versorgungsspannung getrennt werden, kann man erkennen, dass unter Umständen eine erhöhte dynamische Leistung für Einsparungen im statischen Bereich in Kauf genommen wird.

Vor allem für zukünftige Entwicklungen liegt die Abstraktionsebene der Einsparungsmethoden zunehmend höher. Beispielsweise seien hier Power Aware Software und heterogene parallele Verarbeitung genannt. Diese Ansätze zielen darauf ab, in hohen Abstraktionsschichten Entscheidungen zu treffen, Probleme aufzuteilen und mit spezifischer Hardware zu lösen. Allen diesen Vorschlägen ist gemein, dass sie ein Abwägen zwischen Performance, Leistungsaufnahme, Entwurfs- sowie Herstellungskosten und Zuverlässigkeit darstellen. Um in diesem Rahmen die geeigneten Methoden für den jeweiligen Zweck zu finden ist eine Entwurfsraumanalyse unabdingbar.

Die in der Tabelle 2.1 von der ITRS genannten Maßnahmen erheben keinen Anspruch auf Vollständigkeit. Weitere Techniken umfassen z. B. das temporäre Abschalten des Taktsignals mittels Clock Gating oder das Einsetzen mehrerer Betriebsspannungen und verschiedener Schwellspannungen durch Anpassung der Oxiddicke.

Weiterhin gibt es umfangreiche Methoden auf Systemebene, die die genannten Methoden niedrigerer Ebenen anwenden und steuern. Dynamic Voltage and Frequency Scaling (DVFS) wird beispielsweise in heutigen Prozessoren von einer Power-Management-Einheit betrieben, die unter anderem bestimmte Prozessorkerne schneller laufen lässt, wenn die anderen Kerne inaktiv sind (Intel Speedstep [Int04] und Turbo Boost [Pop09], AMD Cool'n'Quiet und Turbo Core). Dies ermöglicht das erlaubte Leistungsbudget des gesamten Chips – bei Auftreten einer einzelnen Aufgabe – besser auszunutzen. Die inaktiven Rechenkerne des Prozessors werden auch als Dark Silicon bezeichnet [EBSA<sup>+</sup>11].

Im Laufe dieser Arbeit werden von den genannten Methoden u. a. symmetrische und heterogene parallele Verarbeitung, HW/SW-Partitionierung, Power and Temperature Awareness und DVFS von besonderem Interesse sein.

## 2.3 Zuverlässigkeit und Robustheit

Gemeinhin wird die Zuverlässigkeit eines technischen Systems als die Fähigkeit angesehen, die geforderte Funktion unter festgelegten Bedingungen für einen festgelegten

Tabelle 2.1: Methoden zur Leistungsreduktion laut ITRS, das Jahr der (voraussichtlichen) industriellen Verbreitung und der Einspareffekt auf dynamische und statische Leistungsaufnahme [ITR12]

		dynamisch	statisch
Low Power Libraries		1,5	1,5
Adaptive Body Biasing		1,2	2
Power Gating		0,9	10
DVFS		1,5	1
Multilevel Cache Architecture	vor 2011	1	1,2
Hardware Multithreading		1	1,3
Hardware Virtualization		1	1,2
Superscalar Architecture		1	2
Symmetric Multiple Processing		1,5	1
Software Virtual Prototype	2011	1,23	1,2
Frequency Islands	2013	1,26	1
Near-Threshold Computing	2015	1,23	0,8
Hardware/Software Co-Partitioning	2017	1,18	1
Heterog. Parallel Processing	2019	1,18	1
Many Core Software Dev. Tools	2021	1,2	1
Power-Aware Software	2023	1,21	1
Asynchronous Design	2025	1,21	1



Zeitraum zu erfüllen [IEE90]. Das Gegenteil des Vorhandenseins von Zuverlässigkeit entspricht einem Systemausfall. Die Robustheit hingegen beschreibt den Grad, zu dem das System seiner korrekten Funktion nachkommt, obwohl es unter dem Einfluss von ungünstigen Eingaben oder schädigenden Umwelteinflüssen steht.

Das Auftreten von Fehlfunktionen gefährdet die Zuverlässigkeit eines Systems allgemein und einer mikroelektronischen Schaltung im Speziellen. Fehlfunktionen und letztlich Ausfälle äußern sich hier beispielsweise durch falsche Befehlssprünge der CPU, fehlerhafte Rechenergebnisse oder unvorhergesehene bzw. ungünstige Systemzustände. Diese Fehlfunktionen werden durch darunterliegende Fehler in der Schaltung hervorgerufen. Dabei können diese Fehler aber auch ohne negativen Effekt und damit transparent bleiben, wenn beispielsweise ein ungünstiger Sprung im Programm-Code trotzdem zum richtigen Endergebnis führt.

Die den Fehlfunktionen zugrundeliegenden Fehler können in digitalen Systemen als einfache Bitfehler interpretiert werden. Dies führt zum X-Modell der Fehlermodellierung (siehe Abbildung 2.3), bei dem das fehlerhafte Bit als zentraler Ansatzpunkt gewählt wird [HBB<sup>+</sup>11]. Auftretende Bitfehler können weiter systematisiert werden. Ein Bit kann dauerhaft einen bestimmten Wert haben (Stuck-At-Fehler), obwohl es seinen Zustand hätte ändern sollen oder der Zustandswechsel kommt zu spät (Timing-Fehler). Weiterhin ist es möglich, dass Bitfehler einzeln oder als zusammenhängender Fehlervektor an mehreren Stellen auftreten. Falls mehrere Fehler auftreten, kann der Zusammenhang zwischen diesen räumlich oder zeitlich korrelieren. Außerdem können Bitfehler permanent oder als wiederkehrende Fehler sowie vorübergehend als transiente Fehler auftreten.

Die direkte Ursache für das Auftreten von Fehlern sind sogenannte Ausfallmechanismen – die eigentlichen Defekte innerhalb der Schaltung. Diese Mechanismen basieren auf physikalischen Vorgängen innerhalb sowie in der Umgebung der Schaltung und lassen sich in plötzlich auftretend und langsam eintretend einteilen. Elektromigration (EM) und Time Dependent Dielectric Breakdown (TDDB) sind Beispiele für Alterungsvorgänge, die langsam eintreten. Alterungserscheinungen verändern die Eigenschaften einer Schaltung typischerweise graduell, bis der Effekt so weit fortgeschritten ist, dass ein Fehler auftritt. Das plötzliche Auftreten eines Fehlers kann unter anderem durch elektromagnetische Einstrahlungen (EMI) von außen, elektromagnetische Kopplungen innerhalb der Schaltung oder physische Defekte verursacht werden.

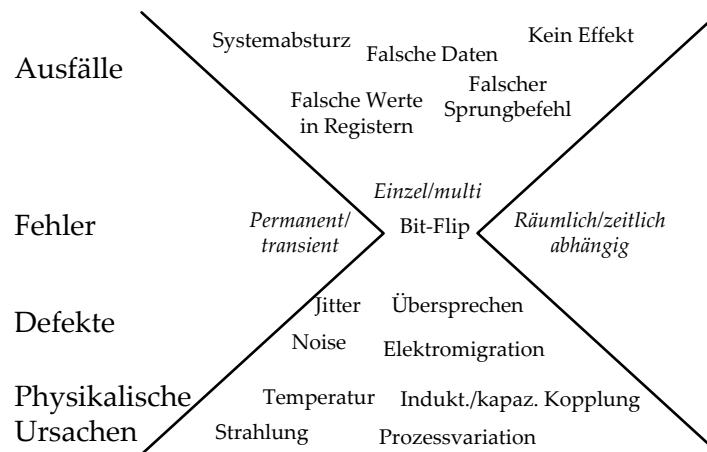


Abbildung 2.3: X-Modell zur Fehlermodellierung [HBB<sup>+</sup>11]

Eine weitere Gruppe von Defekten sind Fehlfunktionen, die schon vor Inbetriebnahme der Schaltung vorhanden waren. Dazu zählen einerseits logische Fehler im Schaltungsentwurf sowie nicht eingehaltene Entwurfsregeln. Andererseits betrifft dies auch Defekte die durch oder während der Produktion entstehen. Gerade die immer anspruchsvolleren Technologien im CMOS-Bereich stellen hohe Ansprüche an die Genauigkeit des Herstellungsprozesses. Hier lässt sich weiter in systematische Prozessunzulänglichkeiten als Ursache und zufällige Variationen der späteren Schaltungsparameter unterteilen.

Alle Fehlfunktionen haben grundsätzliche physische Ursachen. Besonders zu betonen sind hier extreme Betriebsparameter wie hohe mechanische sowie thermische Belastungen, Strahlungsintensität, Feuchtigkeit oder Drücke. Aber auch besonders große oder oft auftretende Schwankungen der physischen Parameter können Ursache für Defekte einer Schaltung sein.

Das angesprochene X-Modell bietet einen guten Anhaltspunkt, den Ansatz für eine Modellierung von Fehlern zu finden. Da alle Defektmechanismen in Bitfehlern münden und alle Fehlfunktionen von fehlerhaften Bits herrühren, liegt der Schluss nahe, diese Ebene für die Modellierung von Zuverlässigkeit und Robustheit zu nutzen.

Es kommt jedoch sehr darauf an, welche spezifischen Parameter des Systems untersucht werden. Während auf der logischen Ebene alle Auswirkungen von Bitfehlern erkannt werden können, ist die Betrachtung von einzelnen Bits in anderen Fällen nicht sinnvoll. Falls beispielsweise die Auswirkungen eines bestimmten physikalischen Umgebungsparameters auf einen Defektmechanismus relevant sind oder die Robustheit einer Software

gegen falsche Code-Sprünge getestet werden soll, kann eine Modellierung mit Hilfe einzelner Bits nicht weiterhelfen. Dann muss eine Betrachtung genau dieses Mechanismus erfolgen. Festhalten sollte man jedoch, dass die Modellierung von Zuverlässigkeit und Robustheit eines logischen Systems nur unter der Berücksichtigung von Bitfehlern möglich ist. Sollten Probleme auf niedrigeren Ebenen untersucht werden, müssen beispielsweise physikalische Modelle genutzt werden. Werden Fehler auf höherer, z. B. algorithmischer Schicht analysiert, ist es eventuell zu aufwändig auf niedrigeren Ebenen zu modellieren.

### 2.3.1 Temperaturabhängigkeit von Ausfallursachen

Typischerweise liegt die Umgebungstemperatur eines elektronischen Systems während des Betriebs etwa bei 20 °C. Es gibt jedoch Fälle, in denen die Umgebungsparameter mehr oder weniger stark abweichen können. Zudem wird die umgesetzte Energie in Wärme umgesetzt. Deshalb muss eine Kühllösung eingesetzt werden, um die Schaltung im gewünschten Temperaturbereich zu halten. Hersteller geben für ihre Produkte meist Temperaturintervalle an, in denen der jeweilige Chip garantiert die gewünschte Funktion erfüllt. Einen Anhaltspunkt gibt die grobe Einteilung in „Commercial“, „Industrial“, und „Military Grade“, die meist für die Intervalle 0 bis 85, -55 bis 125 und -65 bis 175 °C steht.

Je nachdem, welcher physikalische Effekt einen bestimmten Defektmechanismus vorherrschend beeinflusst, wird die Entwicklung mit erhöhter Spannung, erhöhtem Strom, steigenden Temperaturen, Feuchtigkeit oder einer Kombination davon beschleunigt. Die Geschwindigkeit der Entwicklung von Defektmechanismen in mikroelektronischen Schaltungen hängt auf Grund der meist elektro-chemischen Reaktionen von der herrschenden Temperatur ab. Chemische Reaktionen laufen allgemein nach der Reaktionsgeschwindigkeit-Temperatur-Regel (RGT-Regel) ab. Diese Faustformel besagt, dass eine Reaktion bei 10 °C Temperaturänderung doppelt so schnell bzw. langsam abläuft [Röm53]. Diese Regel wurde von Svante Arrhenius zur Arrhenius-Gleichung (siehe Gleichung (2.13)) erweitert [MW97]. Sie beschreibt den quantitativen Zusammenhang zwischen der Reaktionsgeschwindigkeitskonstante ( $k$ ) und der Temperatur. Dabei wird der Frequenzfaktor ( $A$ ) leicht von der Temperatur beeinflusst. Der Exponent der Exponentialfunktion besitzt jedoch einen weitaus größeren Einfluss. Deshalb gilt, dass elektrochemische Reaktionen in mikroelektronischen Systemen exponentiell von der Temperatur abhängen. Die Aktivierungsenergie ( $E_A$ ) bestimmt die konkrete Ausgestaltung des quantitativen Zusammenhangs. In bestimmten Fällen kann sie auch negativ sein und somit eine umgekehrte

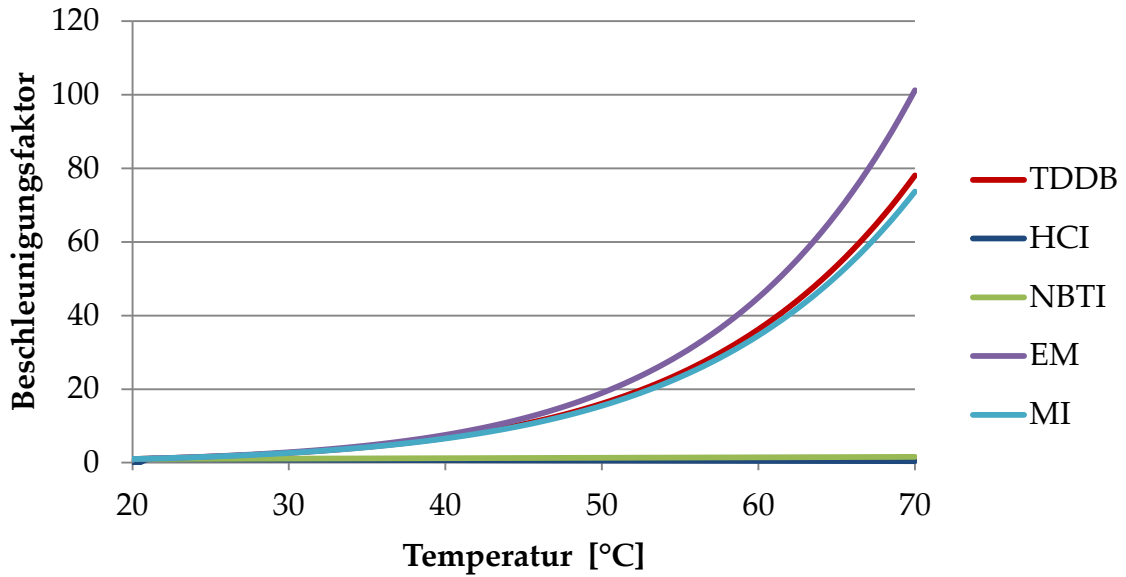


Abbildung 2.4: Temperatureinfluss auf Defektmechanismen [JEDEC10]

Korrelation zwischen Temperatur und Reaktionsgeschwindigkeit darstellen. Die Reaktion lief in diesem Fall unter geringeren Temperaturen schneller ab. Um den Einfluss einer Temperaturerhöhung zu ermitteln werden die Reaktionsgeschwindigkeiten bei verschiedenen Temperaturen dividiert und man erhält einen Beschleunigungsfaktor, um den sich der betrachtete Effekt verstärkt (siehe Gleichung (2.14)).

$$k = A \cdot e^{\frac{-E_A}{k_B \cdot T}} \quad (2.13)$$

$$A_F = \frac{k_1}{k_2} = e^{\frac{-E_A}{k_B} \cdot \left( \frac{1}{T_1} - \frac{1}{T_2} \right)} \quad (2.14)$$

Die momentan in der CMOS-Technologie bedeutendsten Defektmechanismen werden im Folgenden kurz beschrieben sowie ihre Temperaturabhängigkeit dargestellt. Eine Abschätzung des Beschleunigungsfaktors im Bezug zu 20 °C wurde für ausgewählte Defektmechanismen nach Modellen des Joint Electron Device Engineering Council (JEDEC) in Abbildung 2.4 dargestellt.

**Time Dependent Dielectric Breakdown (TDDB):** Das Dielektrikum zwischen Gate, Source und Drain kann durch gefangene Ladungsträger eine leitende Verbindung zwischen Gate und Substrat bilden. Dieser unerwünschte Zustand wird als Gate-Oxid-Breakdown bezeichnet. Außer durch zu hohe Spannungen wird das Oxid auch bei relativ schwachem, anliegendem Feld mit der Zeit durchlässig für Ladungsträger. Zunächst leitet der Transistor immer mehr zwischen Gate und Source, diesen kontinuierlichen, aber nichtlinearen Vorgang nennt man Soft Breakdown. Schließlich bildet sich ein dauerhaft leitender Pfad. Dies führt dazu, dass der betroffene Transistor eine Fehlfunktion aufweist und das Zeitverhalten sowie die logische Funktion des Gatters beeinträchtigt wird. Dies kann zum Fehlverhalten der gesamten Schaltung führen. Das Eintreten des Effektes ist von der Temperatur und der Feldstärke über dem Dielektrikum, also dem Schaltzustand und der Versorgungsspannung abhängig. Je nach konkreter Fertigungstechnologie werden verschiedene Modelle bzw. Aktivierungsenergien angenommen, um den Mechanismus vorherzusagen. Nach dem Beispiel der JEDEC [JEDEC10] beschleunigt sich der Effekt um das 10-Fache für eine Steigerung von 20 auf 45 °C und um das 78-Fache für eine weitere Steigerung um 25 °C (siehe Abbildung 2.4).

Auch das Dielektrikum zwischen den Leitungen einer Ebene kann durch die Verwendung von low-k-Materialien und immer kleinere Leitungsabstände anfällig für TDDB werden [BPP<sup>+</sup>11]. Dies kann auch durch andere Effekte, die die Leitungen beeinflussen, wie z.B. Elektromigration begünstigt werden.

**Hot Carrier Injection (HCI):** Ladungsträger die sich durch den leitenden Kanal eines MOSFET bewegen, können eine so hohe Energie transportieren (Hot Carriers), dass sie mit dem Atomgitter des Oxids interagieren, gefangen werden oder den Halbleiter verlassen können (Leakage). Eine Folge von HCI ist die Erhöhung der Schwellspannung des Transistors. Folglich sinkt die Schaltgeschwindigkeit mit der Zeit ab und kann zum Fehlverhalten der Schaltung führen.

Bei niedrigen Temperaturen gibt es mehr Ladungsträger, die eine große Energie erreichen als bei hohen. Deshalb hat die Temperatur auf HCI einen umgekehrten Einfluss. Bei steigenden Temperaturen wird der Effekt abgeschwächt (siehe Abbildung 2.4). [DGY<sup>+</sup>08]

**Negative Bias Temperature Instability (NBTI):** Bei negativer Gate-Source-Spannung am PMOS-Transistor werden durch unterschiedliche Mechanismen Ladungsträger gefangen. Dies erhöht die Schwellspannung und verringert die Mobilität der Ladungsträger. Wenn die Gate-Source-Spannung abgesenkt wird, ist eine teilweise Erholung zu beobach-

ten. Dabei werden aber nicht alle gefangenen Ladungsträger freigegeben. Vor allem mit neueren technologischen Entwicklungen, wie high-k und Metall-Gates kann bei positiver Gate-Source-Spannung ein analoger Effekt am NMOS-Transistor (Positive Bias Temperature Instability, PBTI) beobachtet werden [LKY<sup>+</sup>08]. Dieser Effekt ist im Gegensatz zu NBTI komplett reversibel. Je nach Modell und Oxiddicke wird von einer nur leichten Temperaturabhängigkeit ausgegangen (siehe Abbildung 2.4).

**Elektromigration (EM):** Bewegte Ladungsträger stoßen mit einer bestimmten Wahrscheinlichkeit auf die Ionen des Metallgitters. Dabei wird ihr Impuls übertragen. Die Ionen bewegen sich daraufhin langsam in die Richtung des Elektronenflusses. Die Folge ist die Abtragung von Metall an bestimmten Orten und Anlagerung an anderen Orten. Der Vorgang geschieht vor allem entlang von Korngrenzen (Aluminium) und Oberflächen (Kupfer). Schließlich kann es zu Kurzschlüssen durch Kontaktierung der Anlagerungen und Unterbrechungen durch Abtragung kommen. Außerdem kann sich durch die Änderung der Kapazitäten und Leitfähigkeiten vor dem vollständigen Ausfall des Leiters das Zeitverhalten der Signalübertragung verschlechtern.

Das bekannteste Modell der EM bildet die Blacksche Gleichung [Bla69]. Sie findet mit leichten Modifikationen immer noch Anwendung und enthält neben dem Einfluss der Stromdichte den bekannten Arrhenius-Term für die Abhängigkeit von der Temperatur. Der Einfluss der EM auf die Schaltungszuverlässigkeit ist durch die immer dünner werdenden Leitungen stark gestiegen. Laut JEDEC-Beispiel ist der Einfluss der Temperatur noch größer als der auf TDDB (siehe Abbildung 2.4).

**Surface Inversion durch bewegliche Ionen (Mobile Ions, MI):** Bewegliche Ionen der Alkalimetalle, wie Kalium, Lithium und Natrium, die durch die Produktionsprozesse einzeln im Halbleiter vorkommen, können im elektrischen Feld zur Kathode driften und sich dort anhäufen. An der Grenze zwischen Substrat und Oxid führt dies unter Umständen zu einem leitenden Kanal. Die Folge ist ein erhöhter Leckstrom und die Begünstigung von anderen Mechanismen wie TDDB. Die exponentielle Temperaturabhängigkeit (siehe Abbildung 2.4) erklärt sich durch die mit der Temperatur steigende Beweglichkeit der Ionen.

Auch für andere chemische Vorgänge wie Korrosion, intermetallische Reaktionen und Oxidation gilt arrheniussches Verhalten. Es gilt festzuhalten, dass die meisten der wichtigen Ausfallmechanismen mit steigender Temperatur exponentiell zunehmen.

Nicht direkt von der Temperatur abhängige Ausfallmechanismen sind beispielsweise elektrostatische Entladung und äußere Strahlung.

Festzuhalten ist, dass die Mechanismen fast alle nach der Arrhenius-Gleichung von der Temperatur abhängen. Deren Einfluss ist durchaus unterschiedlich, aber meist steigt mit der Temperatur auch die Geschwindigkeit des Effektes. EM, TDDDB und MI wurden als am meisten von der Temperatur beeinflusste Effekte identifiziert [JEDEC10].

### 2.3.2 Methoden zur Steigerung der Robustheit

Auf allen Entwicklungsebenen bieten sich Möglichkeiten die Zuverlässigkeit eines mikroelektronischen Systems zu erhöhen. Dabei besteht eine Möglichkeit in der Steigerung der Robustheit. Das bedeutet, das System wird gegen auftretende Fehler unempfindlich gemacht. Die Ausgestaltungsmöglichkeiten sind jeweils vielseitig.

Eine Klassifizierung der gängigen Methoden (siehe Abbildung 2.5) kann mit der Einordnung in Fehlervermeidungs- und Beseitigungsstrategien sowie Fehlertoleranz beginnen. Alle diese Techniken benötigen eine Methode, Fehler zu erkennen. Genutzt werden hier eine Vielzahl von Methoden auf unterschiedlichen Ebenen. Bestimmte Codes bilden eine einfache Möglichkeit, unter Nutzung von Datenredundanz Fehlererkennung zu ermöglichen. Die Mehrfachauslegung von Hardwaremodulen und das abschließende Mehrheitsvotum bilden eine weitere Variante. Watchdog- oder Checkpointing-Mechanismen dienen dazu, den vorhergesehenen Programmablauf zu überprüfen. Beim Design integrierte Funktionen zum Testen einzelner Funktionalitäten werden als Built-in Self Tests bezeichnet.

Fehlervermeidung zielt darauf ab, von vornherein – vor allem durch die Auslegung im Technologie- und Fertigungsbereich sowie den Entwurf der Schaltungstechnik und das Layout der Schaltung – wenig Defektpotential zu schaffen. Beispiele sind hier die Verbreiterung von bestimmten Leitungen, die vermutlich später eine hohe Stromdichte führen werden oder die Vervielfachung von Kontaktierungen, falls einige davon nach der Fertigung nicht voll funktionsfähig sein werden.

Eine weitere Möglichkeit außerhalb des Betriebs des Systems bietet sich durch die Beseitigung oder das Umgehen von vorhandenen Schwächen. Durch das Ändern der Substratvorspannung oder das Unterteilen der Menge gefertigter Chips in verschiedene Leistungskategorien und damit Produkte (Binning) wird vor allem die Ausbeute des Fertigungsprozesses erhöht. Da die vorher nur knapp oder nicht erreichten Spezifikationen

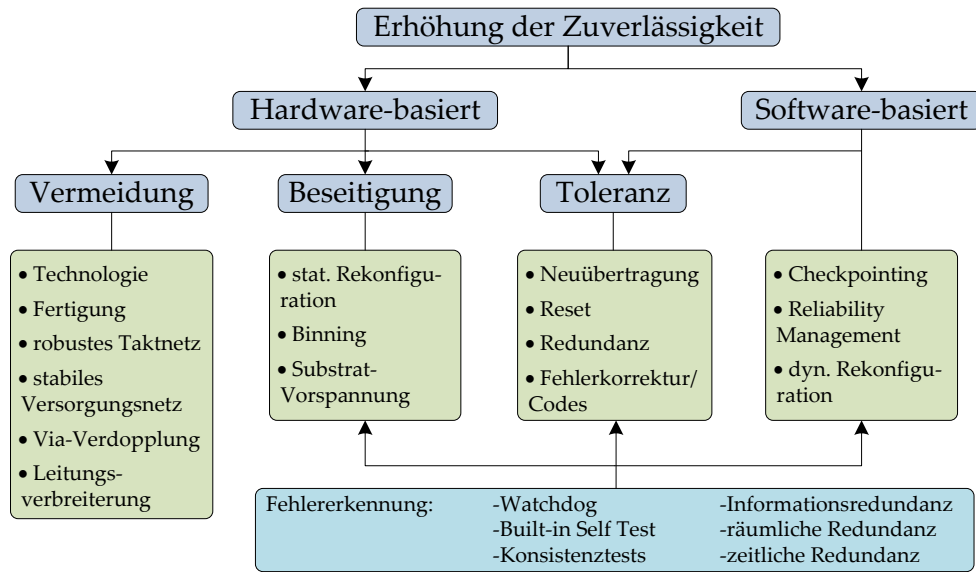


Abbildung 2.5: Methoden zur Steigerung der Zuverlässigkeit [Cor09]

mit den neuen Betriebsparametern aber im beabsichtigten Bereich liegen, erhöht sich auch die Zuverlässigkeit.

Die Maßnahmen während des Betriebs des Systems zielen vor allem auf die Erhöhung der Robustheit. Hier wird mit Methoden der Fehlererkennung und -beseitigung das korrekte Systemverhalten sichergestellt, obwohl temporäre oder dauerhafte Fehlfunktionen auftreten. Genutzt wird hier meist das Prinzip der Redundanz. Die zeitliche oder räumliche Vervielfachung der Informationen wird in Verbindung mit Codes genutzt, um Fehler zu detektieren und zu korrigieren. Während diese Prinzipien auf unterschiedlichen Ebenen im Bereich der Logikfunktion und der Datenübertragung auf Hardwareseite angewandt werden können, gibt es auch Möglichkeiten diese Techniken auf höheren Ebenen mit Software zu nutzen. Neben Techniken zur Fehlererkennung mit Hilfe von Konsistenzprüfungen, Codes oder Checkpointing ergeben sich auch Möglichkeiten das gesamte System mit dem Ziel der Zuverlässigkeit zu rekonfigurieren. Falls beispielsweise eine Systemkomponente fehlerhaft ist, werden Aufgaben auf fehlerfreie Module umgeleitet. Eine Reduzierung der Betriebsfrequenz oder Erhöhung der Betriebsspannung kann ebenfalls dazu dienen, dass fehlerhaft arbeitende Komponenten weiterhin genutzt werden können. Diese Strategie des so genannten „Derating“ wird in bestimmten Situationen auch dazu eingesetzt, eine Alterung oder Überlastung der Schaltung von vornherein



zu verhindern oder unwahrscheinlicher zu machen. Diese Technik, die die Performance deutlich einschränkt, kommt besonders in sicherheitskritischen Bereichen wie Raumfahrt und Militär zum Einsatz.

Generell sind alle hier genannten Ansätze mit Mehraufwand und Kosten verbunden. Seien es Entwicklungskosten in der Fertigungstechnik oder im Entwurf der Schaltung und der Systemfunktionen. Aus diesem Grund gilt es in jedem Fall eine Abwägung zwischen nötigen Maßnahmen zur Erhöhung der Zuverlässigkeit und gebilligter Einbuße an Performance oder letztlich Gewinn zu treffen.

## 2.4 Entwurf

Eine wichtige Folge des rapiden Anstiegs der Integrationsdichte sind die durch die gestiegene Komponentenzahl pro Chip erhöhte Entwurfsfreiheit sowie ein Mehr an Funktionsumfang. Dies bedeutet, dass die Integration von immer mehr funktionalen Einheiten in ein System mit einem erhöhten Arbeitsaufwand einhergeht. Da es zwar Fortschritte in der technologischen Fertigung gibt, der Entwicklungsaufwand jedoch dadurch steigt, entsteht eine Lücke zwischen Produktivität im Systementwurf und dem eigentlich vorhandenen Potential eines Chips. Dieses Phänomen wird auch als Design-Productivity-Gap (DPG) bezeichnet [ITR12].

In Abbildung 2.6 ist die Produktivität der Softwareentwicklung dargestellt, wonach diese am langsamsten steigt. Die Steigerung der Produktivität im Hardwareentwurf konnte nur aufrecht erhalten werden, da die Nutzung von IP-Cores, Many-Core-Systemen und On-Chip-Speicher stark zugenommen hat. Zwischen dieser Kurve und der Steigerung der technologischen Möglichkeiten nach Moore besteht jedoch trotzdem weiterhin der DPG (1). Nach [ITR12] wird außerdem eine noch größere Produktivitätssteigerung in der Softwareentwicklung benötigt, so dass sich neben dem DPG zudem ein Hardware-Software-Gap (HSG) (2) bildet. Dieser sagt aus, dass die Software für die entstehenden Hardwaresysteme nicht mehr effektiv entwickelt werden kann.

Die Leistungsfähigkeit im Entwurf mikroelektronischer Systeme muss daher ständig effizienter werden, um mit den gestiegenen Ansprüchen mitzuhalten. Ein großer Anteil der Effizienzsteigerung wird durch die Verbesserung von unterstützender Software beigesteuert. Die Automatisierung der meisten Schritte – vor allem in den unteren Schichten des Entwurfsablaufs – bildet hier den entscheidenden Ansatzpunkt. Die Automatisierung

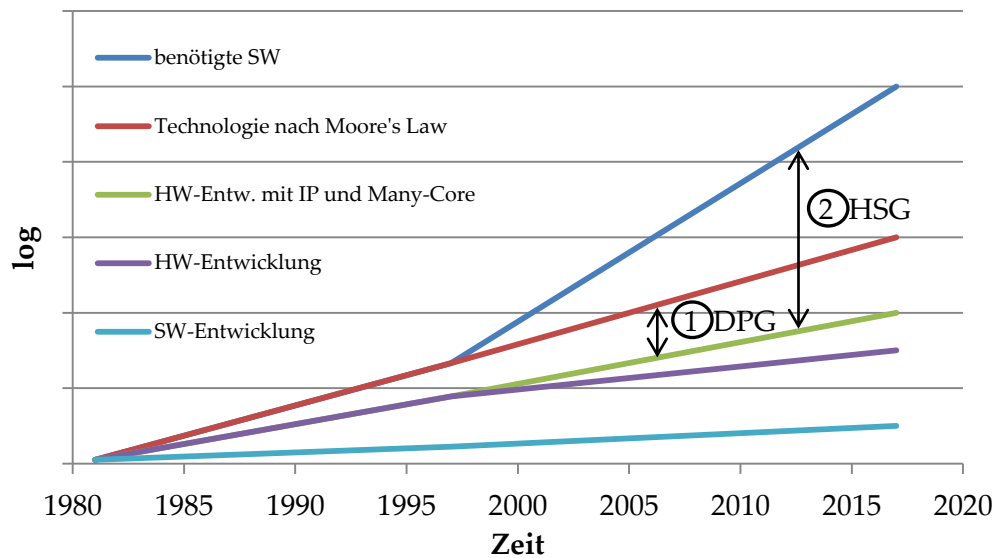


Abbildung 2.6: Hardware-Software-Gap und Design-Productivity-Gap [ITR12]

des Entwurfsablaufes ist jedoch nur bis zu einem bestimmten Grad nützlich, da diese bei intensiver Ausführung in Flexibilität sowie Qualität und späterer Performance nicht mit einer manuellen Optimierung mithalten kann.

Eine weitere Verbesserung in diesem Bereich liegt in der Modularisierung des Systems. Die Schnittstellen zwischen den einzelnen Teilen müssen insofern standardisiert sein, dass diese sich einzeln entwickeln und später zum Gesamtsystem zusammenfügen lassen. Auch Wieder- und Mehrfachverwendung wird so vereinfacht. Gerade reguläre Systemarchitekturen durch die Verwendung homogener Strukturen lassen sich auf diese Weise schneller entwickeln. Dadurch, dass Standard-Schnittstellen Verwendung finden, müssen nicht alle Komponenten von ein und demselben Unternehmen entwickelt werden. Einzelne Teile des Systems werden von spezialisierten Teilnehmern eines Marktes entwickelt und angeboten. Die entstehenden Systemkomponenten werden auch als Intellectual Property (IP) bezeichnet.

Ein weiteres Problem ergibt sich in der Softwareentwicklung für die erstellten Systeme. Die Produktivität steigert sich hier so wie in der Hardwareentwicklung langsamer als die technologische Leistungsfähigkeit. Der Bedarf an Software, die die gesteigerten Fähigkeiten der Multi- und Many-Core-Chips ausnutzen kann, steigt dadurch an. Die

Softwareentwicklung bedient sich angesichts dieses Problems gewöhnlich mehrerer Abstraktionsebenen. Es werden immer mehr Schichten in den Softwareentwurf eingezogen, um von dem eigentlichen System unabhängig zu werden. Die zur Verfügung gestellte Performance kann ab einem bestimmten Punkt jedoch nur noch in geringem Maße genutzt werden, da jede zusätzliche Abstraktionsschicht einen gewissen Effizienzverlust zur Folge hat.

Das Phänomen der fehlenden Effizienzsteigerung des Softwareentwurfs wird auch als Software-Design-Gap im Gegensatz zum vorher beschriebenen Hardware-Design-Gap beschrieben. Ein Ansatz, der beide Probleme zu mindern versucht, ist die Modellierung auf Systemebene zu einem möglichst frühen Zeitpunkt im Entwurfsablauf. Die Funktionalitäten des Systems werden grob algorithmisch beschrieben und die Kommunikation wird in hoher Abstraktionsstufe beschrieben. Das entstehende Modell wird so weit verfeinert, bis eine Simulation der Software und der Interaktion der einzelnen Module möglich ist. Diese Referenz für alle weiteren Implementierungsschritte wird als goldenes Modell bezeichnet [Ghe06]. Der große Nutzen für die Verringerung des DPG ist die Möglichkeit das Referenzmodell für den Softwareentwurf zu nutzen, während die eigentliche Hardware weder physikalisch noch als ausdifferenzierter Entwurf vorhanden ist.

## 2.5 Kommunikationsarchitekturen – Networks-on-Chip

Die Entwicklung hin zu einer größeren Zahl vollständiger Module auf einem Chip stellt erhöhte Anforderungen an die Kommunikation zwischen diesen Einheiten. Eine Möglichkeit, dem gerecht zu werden, besteht darin, jedes Modul mit jedem anderen zu verbinden, mit dem Daten ausgetauscht werden. Diese Punkt-zu-Punkt-Verbindung (siehe Abbildung 2.7(a)) bietet eine hohe und garantierte Bandbreite, sowie eine festgelegte Übertragungslatenz. Bei aktuellen Multi-Core-Prozessoren sind die CPUs aus diesen Performancegründen noch direkt untereinander verbunden – z. B. Intels Quick Path Interconnect (QPI). Während prinzipiell immer mehr Verbindungen hinzugefügt werden können, steigt die Komplexität des Netzes mit dem hinzukommenden potentiellen Kommunikationspartner stark an. Dies führt zu aufwändigen Strukturen, die nicht mehr kosteneffizient realisiert werden können (siehe Abbildung 2.8). Aus diesen Gründen reicht diese Art des Vernetzens für moderne SoCs und Many-Core-Systeme bei Weitem nicht aus.

Zunächst geht man diesem Problem mit dem Einsatz von Bussystemen (siehe Abbildung 2.7(b)) aus dem Weg. Hier sind alle Kommunikationsteilnehmer über ein geteiltes Medium miteinander verbunden. Wenn eine Datenübertragung erfolgen soll, muss mit Hilfe von Arbiterschaltungen entschieden werden, welcher Teilnehmer senden darf. Eine Adressierung ermöglicht die Auswahl des richtigen Empfängers. Mit einem Bussystem ist es sehr einfach neue Module an das Kommunikationssystem anzuschließen. Dieser zentrale Kommunikationsansatz hat jedoch den Nachteil, dass das Medium unter allen Teilnehmern geteilt wird. Dies sorgt für eine nicht immer vorhersagbare Latenz, begrenzt die Bandbreite und schafft somit einen Flaschenhals, den man beispielsweise mit dem Einsatz von geteilten (Ring-)Bussen begegnen kann. Hier wird der Bus in Sektionen unterteilt, die einzeln arbitriert werden können. Auf diese Weise können Teilnehmer in unterschiedlichen Sektionen unter Umständen gleichzeitig kommunizieren. Dies steigert die Gesamtbandbreite. Die Skalierbarkeit ist jedoch weiterhin eingeschränkt.

Einen ähnlichen Ansatz bieten hierarchische Bussysteme (siehe Abbildung 2.7(c)), wie sie z. B. in aktuellen Multimedia-SoCs eingesetzt werden. Einzelne Gruppen von Cores werden intern jeweils mit einem Bus verbunden, während zwischen diesen Gruppen einzelne Verbindungen (Bridges) bestehen, die wiederum die Busse miteinander verbinden. Dieser Ansatz bietet einen großen Gestaltungsspielraum. Man kann über die Einteilung der Gruppen sowie die Auswahl und Gestaltung der einzelnen Busse eine sehr feinteilige Entwurfsraumanalyse durchführen und die Implementierungskosten gut gegen die benötigte Performance abwägen.

Die hierdurch beginnende Entwicklung führt zu einer größeren Vermaschung und letztlich zu einem Verbindungsnetzwerk. Im Falle von einem einfachen hierarchischen Bus kann man noch nicht von einem NoC sprechen. Fügt man jedoch mehr Bridges hinzu und implementiert ein passendes Routing-Prinzip mit Paketvermittlung, ist der Weg zum NoC geebnet.

Das NoC (siehe Abbildung 2.7(d)) besteht grundsätzlich aus zwei Bauteilen – dem Router und dem Link. Außerdem wird die Anbindung des IP-Core an den Router oft hinzugezählt und als Network Interface (NI) bezeichnet. Ein solches NI ist allerdings auch in allen anderen Kommunikationsarchitekturen nötig. Die Links verbinden die NI mit einem Router oder mehrere Router untereinander und bestehen im Wesentlichen aus parallelen Leitungen sowie eventuell aus Performancegründen nötigen Registerstufen und Treibern. Die Router bilden die Knotenpunkte des NoC. Im Extremfall sind alle IP-Cores

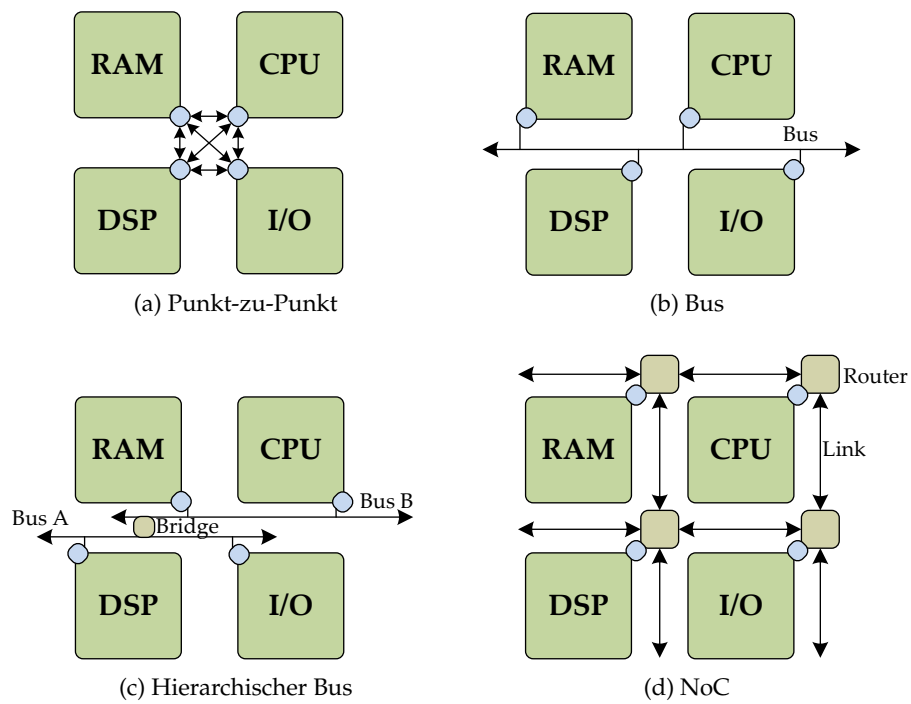


Abbildung 2.7: Übersicht verschiedener Kommunikationsarchitekturen

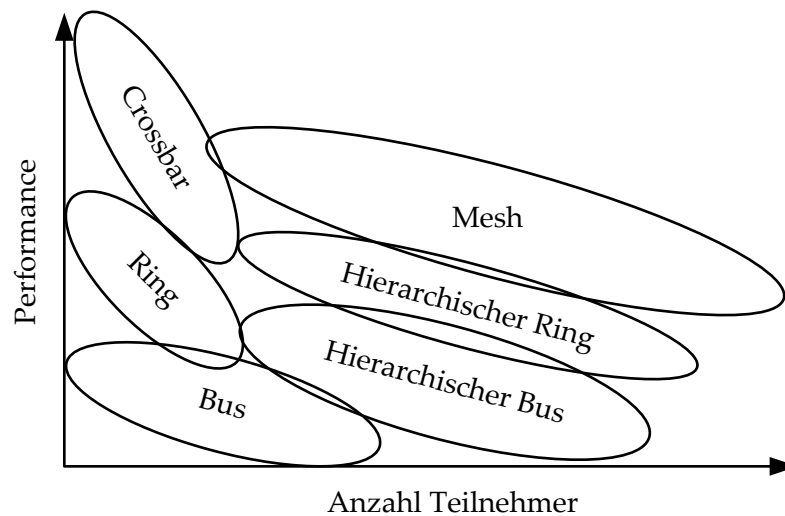


Abbildung 2.8: Leistungsvergleich verschiedener Kommunikationsarchitekturen – Performance als Produkt aus hohem Durchsatz, niedriger Latenz, Verhalten bei Bursts und weit verteilten Zielen [Sal09]

an einen Router angeschlossen. Diese als Single-Crossbar bezeichnete Architektur macht die Vorteile des NoC jedoch nur schlecht sichtbar. Das Hinzufügen von untereinander verbundenen Routern führt zu einem vermaschten Netz. Neben dem höheren Ressourcenverbrauch (Chipfläche) und der unter Umständen nicht deterministischen Latenz als Nachteile werden die im Folgenden erläuterten Vorteile evident.

Ein NoC zeichnet sich vor allem durch eine größere Übertragungsbandbreite aus, da viele Datenströme zeitlich sowie räumlich parallel durch das Netzwerk fließen können. Die topologische Gestaltung des Netzes kann viele Formen annehmen. Zu unterscheiden sind irreguläre und reguläre Formen. Wobei erstere für heterogene SoCs geeignet sind und eine angepasste Performance liefern, jedoch Nachteile in der Erweiterbarkeit besitzen. Letztere eignen sich eher für homogene Many-Core-Systeme und zeichnen sich durch ihre bessere Skalierbarkeit aus. Als Beispiele für die hier betrachteten regulären Topologien können 2D- und 3D-Gitternetze (Mesh), erweiterte Mesh-Strukturen, wie Border Enhanced Mesh (BEAM) oder Torus-Formen betrachtet werden. Allgemein gesehen haben NoCs eine sehr gute Skalierbarkeit, da die Gesamtbandbreite des Systems durch Hinzufügen von Routern und Links linear steigt.

Ein weiterer Vorteil ist die gute Integrierbarkeit von verschiedenen IP-Cores und deren Wiederverwendbarkeit. Die Teile des NoC selbst lassen sich durch die reguläre Struktur, also die Gleichheit der grundsätzlichen Bausteine, ebenfalls effizient implementieren. Diese Vorteile machen die Verwendung von NoCs gerade in Anbetracht des DPG attraktiv. Ab einer gewissen Anzahl an Teilnehmern bietet das NoC entscheidende Performance-Vorteile gegenüber den anderen genannten Kommunikationsarchitekturen (siehe Abbildung 2.8).

Eine detailliertere Betrachtung der einzelnen Teile eines NoC und der zu treffenden Design-Entscheidungen wird in den folgenden Kapiteln, insbesondere Abschnitt 4.1 geliefert.

### 2.5.1 Forschung

Während die grundsätzliche Idee, dass NoCs die Zukunft der On-Chip-Kommunikationsarchitekturen sein werden, sich bereits in den 1990er Jahren aufdrängte, erfolgte 2002 von Benini und De Micheli [BDM02] eine vielzitierte Veröffentlichung, die das NoC als neues SoC-Paradigma darstellte. Seitdem erfreut sich die Forschung an diesem Thema einer steigenden Beachtung durch die Wissenschaft (siehe Abbildung 2.9). Im Jahr 2010 scheint zunächst ein Höhepunkt der Anzahl wissenschaftlicher Veröffentlichungen mit dem Hauptthema NoC erreicht zu sein. Seitdem stagniert die Zahl der jährlich veröffentlichten Beiträge. Anzumerken ist, dass das Jahr 2013 zu diesem Zeitpunkt (Mai 2014) noch nicht vollständig vom IEEE Xplore Index erfasst wurde.

Bei Berücksichtigung des Faktes, dass wissenschaftliche Veröffentlichungen oft erst ein oder zwei Jahre nach dem eigentlichen Beginn der Forschungen erfolgen, kann man teilweise die Entwicklung nach dem Gartner-Hype-Cycle ablesen (siehe Abbildung 2.10), obwohl dieser auf die wirtschaftliche Entwicklung einer Technologie ausgelegt ist. Die etwas überzogenen Erwartungen wurden hier z. B. in den Jahren um 2007 gesehen. Eine Abschwächung des Interesses wird in der Phase „Ernüchterung“ sichtbar, die im Jahr 2010 gesehen wird. Nach 2010 hat auch die Zahl der wissenschaftlichen Veröffentlichungen leicht abgenommen. Letztendlich wird das NoC-Konzept im Jahre 2013 als produktiv einsetzbar und damit auch wirtschaftlich relevant eingeschätzt. Dies passt auch zu den ersten kommerziellen Produkten, die im Jahr 2012 und 2013 damit warben, auf einem NoC zu basieren. Weiterhin haben einige Unternehmen sich im Rahmen von Entwicklungsprojekten in Form von angewandter Forschung am Thema NoC beteiligt (siehe Kapitel 2.5.2).

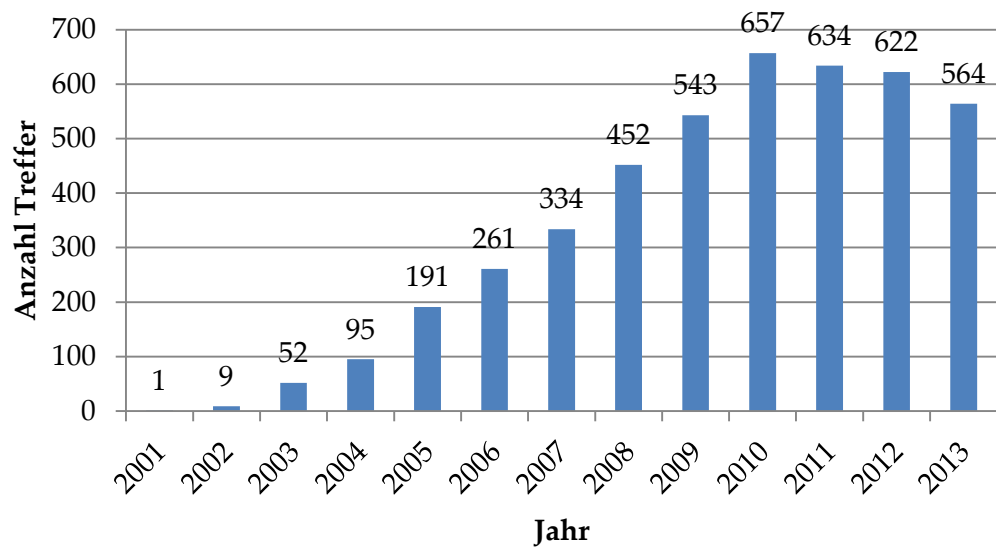


Abbildung 2.9: Jährliche Anzahl der Suchergebnisse in Zusammenfassung und Metadaten von Konferenzbeiträgen, Zeitschriften und Büchern bei IEEE Xplore (Suchbegriff: Network-on-Chip, Networks-on-Chip oder NoC)

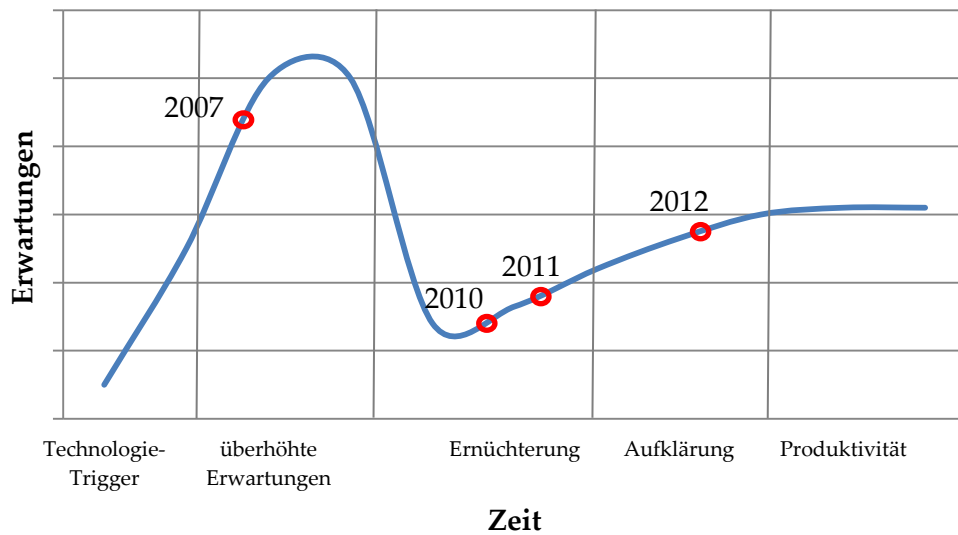


Abbildung 2.10: Gartner Hype Cycle für Networks-on-Chip [Shu12]



Wissenschaftliche Betrachtungen gibt es, wie erwähnt, zahlreiche. Zu den bekannteren Beispielen zählt die RAW-Architektur des MIT [TKM<sup>+</sup>02] – einer Mesh-Topologie mit zunächst 16 Kacheln, bestehend aus je einem einfachen RISC-Core und einem 5-Port-Router. Ein weiteres Beispiel bildet das in Stockholm entwickelte NOSTRUM [J<sup>+</sup>00], welches eine sehr ähnliche Architektur aufweist. In dem Entwurf Æthereal [GDR05] wird mehr Wert auf eine Datenübertragung unter Einhaltung bestimmter Garantien, wie Übertragungszeit, Durchsatz und Reihenfolge gelegt. Ebenfalls mehr auf Quality of Service (QoS) konzentriert sich die Entwicklung von QNoC [BCGK04]. Weitere NoCs als Basis wissenschaftlicher Betrachtungen sind z. B. in [SKH08] aufgeführt.

### 2.5.2 Anwendung

Hauptsächlich durch universitäre Ausgründungen sind NoCs in Many-Core-Form seit kurzem auch kommerziell verfügbar. Das System Raw des MIT [TKM<sup>+</sup>02] – zunächst ein Testchip mit 16 Cores – wurde beispielsweise von dem Unternehmen Tilera zum Produkt Tile-Gx weiterentwickelt. Tilera nennt das Kommunikationssystem seines SoC iMesh. Es besteht momentan aus bis zu 72 Routern in 2D-Mesh-Topologie. Das SoC beherbergt neben generell nutzbaren Prozessoren (General Purpose Processors – GPP) auch Speicher- und Netzwerkcontroller sowie Cores zur Verschlüsselung und Datenkompression. Die Cache-Kohärenz innerhalb des Systems wird nicht sichergestellt. Sollten mehrere Cores gemeinsamen bzw. geteilten Speicher nutzen, müssen zusätzliche Maßnahmen auf Softwareebene getroffen werden [Til13].

Das französische Unternehmen Kalray, das aus Aktivitäten der Forschungszentren in Paris und Grenoble hervorging kündigte einen Many-Core-Chip an, der bis zu 256 Rechenkern sowie zusätzlich Speicher-, Netzwerk- und PCIe-Controller enthalten soll. Jeweils 16 Cores bilden einen Cluster. Innerhalb dieser Cluster wird ein Speicher geteilt. Damit ist auch nur innerhalb des Clusters die Cache-Kohärenz sichergestellt. Die Cluster sind untereinander per NoC verbunden. Zu jedem solcher Cluster gibt es einen Router [Kal14].

Das amerikanische Unternehmen Adapteva hat ebenfalls ein Many-Core-System angekündigt, welches ein NoC und zunächst bis zu 64 RISC-Prozessoren enthält. Das Epiphany genannte System enthält ebenfalls ein Mesh-Netzwerk, welches für die Kommunikation der Cores untereinander und mit externem Speicher genutzt wird. Um ihre Produkte bekannt zu machen und an möglichst viele Entwickler und Interessierte zu verteilen, nutzte Adapteva ein Crowdfunding-Projekt. Im Zuge dessen wurden eine prototypische 64- und

eine serienreife 16-Core-Version und ein dazu passendes Board im Checkkarten-Format entwickelt. Dieses Produkt wird nun für rund 100 US-Dollar vertrieben [Ada13].

Für irreguläre Chipstrukturen, also heterogene SoCs werden seit längerem von einigen Unternehmen NoC-Lösungen eingesetzt. So versorgt Arteris u.a. Texas Instruments mit NoC-IP für ihre ARM-basierten OMAP-SoCs, die hauptsächlich in Smartphones und Tablets eingesetzt werden. Ein weiterer IP-Hersteller im NoC-Bereich ist Sonics. Von ARM direkt werden verschiedene Kommunikationssysteme zur Integration mit ihren CPUs angeboten, die aber eher als hierarchische Busse bezeichnet werden müssen (z. B. NIC-400).

Auch etablierte CPU-Hersteller wie Intel haben teilweise Entwicklungen aus dem Bereich NoC aufgegriffen und prototypisch umgesetzt. So hat Intel zwei Entwicklungen vorgestellt, die ein NoC im engeren Sinne enthalten. Zum einen den 80-Core „Tera-Scale Research Chip“ [VHR<sup>+</sup>08], der mit dem Codenamen Polaris bezeichnet wird. Zum anderen den 48-Core „Single-Chip Cloud Computer“ (SCC) [HDV<sup>+</sup>11, SJJ<sup>+</sup>11].

Polaris war zunächst eine Intel-interne Entwicklung. Der Chip besteht aus 80 Kacheln und rund 100 Millionen Transistoren. Jede Kachel enthält einen 5-Port-Router und einen (Gleitkomma-)Prozessor. Bei einem Nominaltakt von 3,16 GHz erreicht das System eine Leistung von etwa 1 TFLOPS. Die Leistungsaufnahme beträgt dann ca. 82 W. Es wurde ein 8x10 2D-Mesh als Kommunikationsnetz verwendet, das eine Bandbreite von 2 Terabit/s erreicht. Polaris war nur einigen universitären Kooperationspartnern von Intel zugänglich. Das Interesse war jedoch groß genug, dass der folgende SCC der wissenschaftlichen Gemeinde in einem größeren Rahmen zugänglich gemacht wurde.

Im SCC integrierte Intel größere CPU-Cores, die der Pentium-Architektur (P54C) entsprechen. Das Kommunikationssystem verbindet hier unter der Nutzung von 24 Routern 48 dieser Prozessoren. Zwei CPUs sind jeweils mit einem Router verbunden und teilen sich einen lokalen (L2 Cache) Speicher. Für jede dieser Kacheln kann eine eigene Betriebsfrequenz gewählt werden. Die Versorgungsspannung ist für jede Gruppe, die aus je vier dieser Kacheln besteht, einstellbar. So kann innerhalb des Systems eine Abwägung zwischen Leistungsaufnahme und Performance erfolgen. Außerdem kann dadurch das thermisch begrenzte Leistungsbudget des relativ großen Systems besser ausgenutzt werden.

Im Gegensatz zu herkömmlichen Multi-Cores, wo zwei gleichzeitig laufende Prozesse auf verschiedenen Cores über geteilten Speicher kommunizieren, können die einzel-

nen Prozessoren sich über das NoC Nachrichten schicken. Diese Technik ähnelt einem Rechencluster und dies prägt wohl auch den gewählten Namen. Der SCC wurde einigen Universitäten physisch und mehreren virtuell zugänglich gemacht, um sich an der Forschung im Bereich der Parallelisierbarkeit von Algorithmen und entsprechender Compilerunterstützung zu beteiligen.

In kommerziellen Produkten setzt der langjährige Marktführer Intel jedoch bisher auf andere Kommunikationsstrukturen. In dem vom Larrabee – als Grafik-Prozessor entworfenen – abstammenden Many-Core-Produkt Xeon Phi wird eine doppelte Ringbus-Struktur verwendet. Die bis zu 62 Rechenkerne und mehrere Speichercontroller werden mit den jeweils in eine Richtung laufenden 512 Bit breiten Bussen verknüpft. Diese Ringe können in mehrere kürzere Teilbusse unterteilt werden, falls mehr als 16 Teilnehmer angeschlossen werden. Diese Lösung bietet eine besser vorhersagbare Bandbreite und Latenz als kompliziertere Topologien, die komplexere Routingstrategien benötigen [SCS<sup>+</sup>08].

## 2.6 Implikationen und Ziele der Arbeit

Die Verwendung von NoCs als Kommunikationsarchitektur von Many-Core-Systemen stellt eine notwendige Entwicklung in der Evolution von hoch parallelen Rechensystemen dar. Wie genau das NoC implementiert wird, hat maßgeblichen Einfluss auf die Leistungsfähigkeit, den Energie- und Verlustleistungsumsatz sowie die Zuverlässigkeit. Die auszuwählenden Designparameter können auf diese unterschiedlichen Zielparameter angepasst werden. Den nötigen Kompromiss bzw. das für den Anwendungsfall optimale NoC sollte man während des Entwurfsablaufs möglichst früh finden. Spätere Entscheidungen sowie Änderungen haben Mehrkosten zur Folge und Verzögern die Entwicklung des Systems. Aus diesen Gründen ist eine möglichst schnelle Lösung, mit der unterschiedliche Parameter des Systems bewertet werden können, notwendig.

Im weiteren Verlauf dieser Arbeit soll deshalb am Beispiel eines NoC gezeigt werden, wie eine Entwurfsraumanalyse durchgeführt werden kann. Weiterhin wird untersucht, welche Faktoren speziell in Bezug auf die Leistungsaufnahme eine Rolle spielen. Dabei sollen Ansätze und Lösungen untersucht werden, wie der Analyseaufwand reduziert werden kann, damit der Entwurfsraum möglichst umfangreich geprüft werden kann. Außerdem werden Verkehrssituationen betrachtet, die mit bestimmten Anwendungsfällen korrelieren. Dies soll praxisnahe Szenarien mit Bezug auf die Verkehrssituation im NoC

simulieren. Es werden entsprechende Auswirkungen auf die entstehende Verlustleistung untersucht sowie deren Verteilung und die daraus resultierende Temperatúrausbreitung betrachtet.

Um diese Ziele zu erreichen, müssen zunächst die Grundstrukturen eines NoC untersucht werden. Dabei wird insbesondere Wert auf die Leistungsaufnahme gelegt. Weiterhin ist eine Simulationsumgebung auf Systemebene zu schaffen, die die nötigen Zielparameter ermitteln kann und eine hohe Performance aufweist.

# Kapitel 3

## NoC-Links

Zunächst werden in diesem Kapitel die Grundlagen der Übertragungskanäle zwischen den Routern behandelt. Dabei wird darauf eingegangen, wie diese Links in traditioneller Fertigungstechnologie beschaffen sind und wie sie – vor allem in Hinblick auf die Leistungsaufnahme – modelliert werden können.

Anschließend wird grundlegend auf den Crosstalk-Effekt eingegangen, der vor allem auf parallel verlaufenden Leitungen, wie sie in NoCs vorkommen, auftritt. Daraufhin wird die Modellierung der durch den Effekt entstehenden Datenabhängigkeit betrachtet. Da Crosstalk das Zeitverhalten sowie die Leistungsaufnahme beeinträchtigen kann, wird zum einen – durch eine eigens entworfene Schaltung – die Signalverzögerung innerhalb eines Field Programmable Gate Arrays (FPGA) untersucht [GWWT12]. Zum anderen wird die Datenabhängigkeit der Leistungsaufnahme und des Energieumsatzes mit verschiedenen praktisch vorkommenden Datenmustern bestimmt [GWT10]. Die beiden Abschnitte 3.3.1 und 3.3.2 basieren jeweils auf der oben genannten eigenen Publikation.

### 3.1 Grundlagen

Die Übertragung von Daten innerhalb eines NoC erfolgt über die Links, die die Router miteinander verbinden. In herkömmlichen Technologien werden dafür parallele Leitungen in die Metallebenen der Chips eingebracht. Neuere Techniken, wie optische oder serielle Übertragung der Daten, konnten sich hauptsächlich aufgrund des nötigen Konvertierungsaufwands an Beginn und Ende des Links bisher nicht durchsetzen. Links werden meist

synchron betrieben. Das bedeutet, dass die Daten an Sender sowie Empfänger mit der gleichen Taktrate eingespeist und abgetastet werden. Links entsprechen damit im Wesentlichen kurzen Busabschnitten. Diese bestehen ebenfalls aus einfachen Leiterstücken. Zur Modellierung von Bussen existieren bereits seit einiger Zeit ausführliche wissenschaftliche Betrachtungen. Deren Grundlagen werden im Folgenden aufgeführt.

Eine Leitungsebene in Mikrotechnologie besteht aus leitendem und isolierendem Material. Die Leitungen können als aneinandergefügte quaderförmige Metallstücke modelliert werden. Typischerweise beinhaltet jede Ebene Leitungen, die entweder in x- oder y-Richtung verlaufen. Zwischen den Ebenen liegt eine Isolationsschicht. Sie können über Durchkontaktierungen miteinander verbunden werden. Mehrere Leitungsstücke dieser Art sind in Abbildung 3.1 dargestellt (Breite ( $w$ ), Länge ( $l$ ) und Höhe ( $h$ )). Die Breite sowie der Abstand der Leitungen zueinander ( $s$ ) werden in Vielfachen des minimal möglichen Rasterabstandes ( $\text{pitch}$ ), der durch die Fertigungstechnologie vorgegeben ist, gewählt.

Die funktionalen Eigenschaften der Leitungen werden von ihren elektrischen Grundeigenschaften beeinflusst. Da es sich um passive Schaltungselemente handelt, sind hier zunächst die Widerstände, Kapazitäten und Induktivitäten zu betrachten.

Bis zu Technologien der 250 nm-Generation wurde meist Aluminium für die Leiterbahnen verwendet [SML98] (spezifischer Widerstand  $0,027 \Omega \frac{\text{mm}^2}{\text{cm}}$ ). Im Rahmen der fortschreitenden Miniaturisierung wurde seitdem auf Kupfer ( $0,017 \Omega \frac{\text{mm}^2}{\text{cm}}$ ) zurückgegriffen. Die niedrigeren Widerstände erlauben eine weitere Verringerung des Leiterquerschnitts. Wobei bei der Verwendung von Kupfer zusätzliche Maßnahmen in Form von Diffusionsbarrieren zum Schutz des Siliziums vor chemischen Reaktionen ergriffen werden müssen.

Jede Leitung hat zu allen anderen Elementen kapazitive Beziehungen. Durch Abschirmung kann dies in erster Näherung auf die nächsten Nachbarn eingeschränkt werden. Dies resultiert in einer Modellierung durch je einen Plattenkondensator in alle vier Richtungen. Somit ergibt sich eine Gesamtkapazität aus:

$$C_{\text{gesamt}} = C_g + C_t + 2 \cdot C_c$$

Dabei entspricht  $C_t$  der nach oben und  $C_g$  der nach unten angrenzenden Schicht,  $C_c$  bezeichnet die kapazitive Beziehung zwischen den Leitungen einer Ebene. An den Rändern der Busstruktur liegen die so genannten Fringe-Kapazitäten  $C_f$ . Die Bestimmung dieser Kapazitäten ist unter Einbeziehung der Layout-Informationen und der verwend-

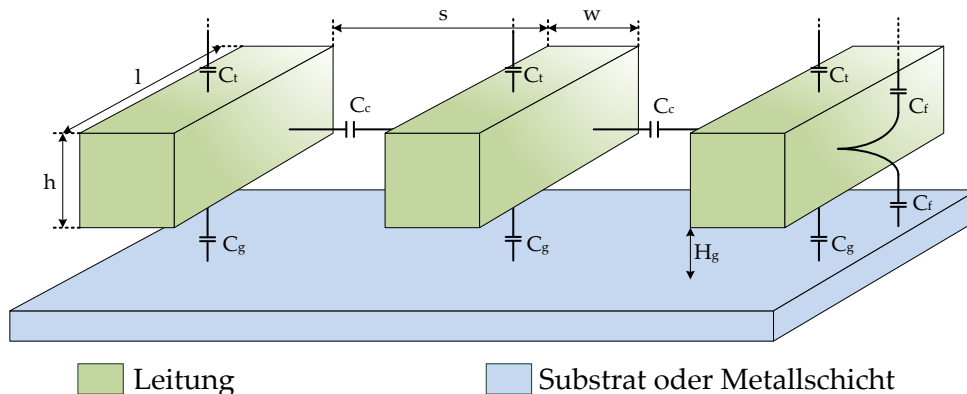


Abbildung 3.1: Schematische Darstellung einer Leitungsebene innerhalb eines Chips

ten Technologie möglich. Die genauesten Ergebnisse werden jedoch durch aufwändigere Integrationsverfahren wie die Finite-Elemente-Methode (FEM) erreicht [MG89]. Während der Layout-Phase arbeiten die EDA-Werkzeuge mit für jede Leitungsebene und die entsprechenden Abstandswerte stückweise hinterlegten Informationen der Technologiebibliotheken. Eine weitere Methode, die ohne die konkrete Technologiebibliothek auskommt, ist die näherungsweise Berechnung nach gelösten Differentialgleichungen [WLM00]. Die Materialkonstanten und geometrischen Dimensionen müssen dafür jedoch trotzdem bekannt sein.

Für die Isolationsschichten zwischen den Leitungen wird seit der Generation von 130 nm [GAA<sup>+</sup>00] Material mit extrem niedriger Permittivität genutzt. Die Schwierigkeit für die Materialwissenschaften besteht darin, die unterschiedlichen elektrischen und mechanischen Parameter abzuwägen. Die elektrische und dielektrische Leitfähigkeit soll möglichst gering sein, während die Haltbarkeit sowie Verarbeitbarkeit hoch sein müssen. Ein Permittivitätswert unter dem von Siliziumdioxid (3,9) wird dabei als low-k bezeichnet. Momentan wird unter anderem daran gearbeitet Luftspalte einzufügen, was die Leitfähigkeiten weiter senken würde.

Die Modellierung von auftretenden Induktivitäten stellt sich komplizierter dar, da die Anzahl der Beziehungen zwischen den Schaltungselementen nicht so stark eingegrenzt werden kann. Einschränkende Regeln während des Schaltungsentwurfs (Design Rules) verhindern einen größeren Einfluss. Die Betrachtung von induktiven Einflüssen

beschränkt sich deshalb meist auf Versorgungsleitungen, analoge Schaltungen und Gehäusebau [Cor09].

Der wichtigste funktionale Parameter einer Leitung ist die Signallaufzeit. Diese ist eng mit weiteren Aspekten wie der Qualität der Signalfanken gekoppelt. Um die durch die Zeitkonstante  $\tau = R \cdot C$  der Leitung gegebene Laufzeit zu reduzieren, werden zusätzliche Treiber eingefügt. Die quadratische Abhängigkeit der Signallaufzeit von der Leitungslänge kann so nahezu linearisiert werden [PD10]. Die zusätzlich eingefügten Treiber werden in CMOS-Technik durch Inverter realisiert. Diese aktiven Elemente erhöhen den Leistungsbedarf der Schaltung und beinhalten eine zusätzliche zeitliche Verzögerung. Diese Aspekte stellen den begrenzenden Faktor für die Wirksamkeit dieser Technik dar.

### 3.2 Abschätzung von Leistungs- und Flächenbedarf

Die dynamische Leistungsaufnahme einer Leitung setzt sich aus der Umladung der beschriebenen Kapazitäten und dem Leistungsbedarf der vorhandenen Treiber zusammen. Sobald die Leitungslänge und die Zieltechnologie bekannt sind, kann abgeschätzt werden, wie viele Treiber benötigt werden und wie diese zu dimensionieren sind, um eine bestimmte Signallaufzeit zu erreichen. Der resultierende Leistungsbedarf steht daraufhin in ausreichender Genauigkeit fest. Eine Ausnahme bilden Einflüsse von außen, zu denen man auch den Crosstalk-Effekt von benachbarten Leitungen zählen kann. Dieser kann den Leistungsbedarf zumindest kurzzeitig erhöhen oder verringern. Eine genauere Evaluierung dieses Einflusses erfolgt in den nächsten Abschnitten. Außerdem ist der Leistungsbedarf von den zu übertragenden Daten abhängig. Aus diesem Grund werden Abschätzungen ohne bekannte Daten oder deren Statistiken mit einer Schaltwahrscheinlichkeit von 50 %, also gleichverteilten Zufallsdaten, durchgeführt. Dies geschieht unter der Annahme, dass häufiger wechselnde Signale über längere Zeiträume nicht vorkommen.

Um die beschriebenen Daten zur physischen Implementierung zu erhalten, werden im Allgemeinen einfache, lineare Modelle genutzt. Dazu kommt die Nutzung verschiedener Algorithmen nach denen die Treiber eingefügt werden [ZMLBYT09, LZKC02, BM02]. Auch DSENT [SCK<sup>+</sup>12] bzw. ORION [KLPS09] und McPAT [LAS<sup>+</sup>13] – Tools zur Vorhersage der Leistungsaufnahme – nutzen dieses Vorgehen.

Der Flächenbedarf ist nicht von den Daten abhängig, weist ansonsten aber prinzipiell die gleiche Vorhersagbarkeit wie der Leistungsbedarf auf. Für beide Abschätzungen



müssen die Zieltechnologie sowie der ungefähre Floorplan des Systems bekannt sein. Mit den beschriebenen Methoden lassen sich Leistungsbedarf und benötigte Fläche von NoC-Links bereits auf hohen Abstraktionsebenen vorhersagen.

### 3.3 Untersuchung des Crosstalk-Effektes

Als Crosstalk oder Übersprechen zwischen induktiv und kapazitiv gekoppelten Leitungen wird die Übertragung von Signalpegeln zwischen diesen bezeichnet. Das eingekoppelte Signal besitzt durch die Übersprechdämpfung in der Regel eine um ein Vielfaches geringere Signalstärke. Induktives Übersprechen wird aus den in Abschnitt 3.1 genannten Gründen nicht betrachtet. Die folgende Untersuchung ist deshalb auf kapazitive Crosstalk-Effekte konzentriert. Für die Bewertung dieser Effekte sind die Kapazitäten einer Leitung zu den horizontal benachbarten Netzen der entscheidende Faktor. Obwohl auch die Beziehungen zu den darüber- und darunterliegenden Leitungen betrachtet werden könnten, wird dies in diesem Fall nicht getan. Die Leitungen in anderen Ebenen verlaufen gemeinhin nicht parallel zu den betrachteten busähnlichen Strukturen. Die kapazitive Kopplung ist hier im Sinne des Crosstalk-Effektes zu vernachlässigen. Sollte es dennoch dazu kommen, dass Leitungen parallel in verschiedenen Ebenen platziert werden, gelten die folgenden Betrachtungen mit niedrigeren Kopplungsfaktoren ebenfalls.

Die beobachtete Leitung gilt als das Opfer und seine Nachbarn als die Aggressoren. Wenn sich der Spannungspegel auf den Aggressoren ändert, kann man dies als eine Änderung der effektiven Kopplungskapazität betrachten. Diese Kapazität erhöht oder verringert sich, je nachdem, ob das Signal der Aggressoren den gleichen oder den entgegengesetzten Pegelwechsel vollzieht wie das Opfer. Als Beispiel schaltet eine Leitung von 0 auf 1 V, die benachbarten Leitungen schalten von 1 auf 0 V. An der Koppelkapazität muss nun nicht der Potentialunterschied von 1 V sondern von 2 V überwunden werden. Dies benötigt doppelt so viele Ladungsträger und somit auch doppelt so viel Energie. Der Ladevorgang benötigt ungefähr die doppelte Zeit. Im umgekehrten Fall, wenn also die Opferleitung in die gleiche Richtung umschaltet wie die Aggressorleitung, ist die effektive Koppelkapazität null. Die Potenziale auf beiden Seiten des Kondensators verändern sich im gleichen Maße und es findet kein Umladen statt. Die Schaltzeit der Leitung verringert sich gegenüber dem Nominalfall unter konstanten Umgebungsbedingungen entsprechend.

Tabelle 3.1: Mögliche MCF einer Leitung beim Schalten von 0 auf 1

$\begin{array}{c c} & i-1 \\ \hline i+1 & \end{array}$	$0 \rightarrow 0$	$0 \rightarrow 1$	$1 \rightarrow 0$	$1 \rightarrow 1$
$0 \rightarrow 0$	2	1	3	2
$0 \rightarrow 1$	1	0	2	1
$1 \rightarrow 0$	3	2	4	3
$1 \rightarrow 1$	2	1	3	2

Die Änderung der effektiven Kapazitäten hat eine Änderung in der Signallaufzeit zur Folge und ist von dem Verhältnis von konstanter zu änderbarer bzw. Boden- zu Koppelkapazität abhängig (siehe Gleichung 3.1).

$$C_{eff} = C_g + \sigma \cdot C_c \quad (3.1)$$

Dabei hängt  $\sigma$  von der Kombination der Schaltrichtungen des rechten und linken Nachbarn der betrachteten Leitung ab und wird als Miller-Coupling-Factor (MCF) [CKK00] bezeichnet. Die verschiedenen möglichen Kombinationen können den Wert der effektiven Kapazität im Vergleich zum statischen Fall, in dem der MCF 2 entspricht, erhöhen oder verringern (siehe Tabelle 3.1). Der MCF kann unter Verwendung der folgenden Gleichung bestimmt werden, wobei  $v_i^f = 1$  ist, wenn der Finalwert der Spannung an der i-ten Leitung  $V_{DD}$  ist und  $v_i^f = 0$ , wenn er  $V_{CC}$  entspricht.  $V_i^i$  steht für den Initialwert dieser Leitung.

$$\sigma = [-1, 2, -1] \cdot \begin{bmatrix} v_{i-1}^f v_{i-1}^i \\ v_i^f - v_i^i \\ v_{i+1}^f - v_{i+1}^i \end{bmatrix} \quad (3.2)$$

Da die dynamische Leistungsaufnahme von der Kapazität abhängt und diese sich durch Crosstalk ändert, resultiert die dynamische Leistungsaufnahme wie in Gleichung 3.3. Dabei ist  $V^i$  die Anfangs- und  $V^f$  die End-Spannung [SC02]. Der Faktor  $\lambda$  hängt von der Fertigungstechnologie und der verwendeten Metalllage ab. Der Wert steigt mit dem Fortschritt der Technologiegenerationen. In 130 nm-Prozessen beträgt  $\lambda$  ungefähr 1-4 [SS05]. Für 45 nm wird z. B. ein Wert von 12,2 angegeben [SWY13].

Tabelle 3.2: Crosstalk-Muster für das Schalten der Opferleitung von 0 auf 1

Musterklasse	$C_{eff}$	Schaltbeispiel	Anzahl mögl. Muster
0C	$C_g + 0 \cdot C_c$	000 → 111	1
1C	$C_g + 1 \cdot C_c$	000 → 011	4
2C	$C_g + 2 \cdot C_c$	101 → 111	6
3C	$C_g + 3 \cdot C_c$	100 → 010	4
4C	$C_g + 4 \cdot C_c$	101 → 010	1

$$P_{dyn} = a \cdot f \cdot V_i^f \cdot [-\lambda, 1 + 2\lambda, -\lambda] \cdot \begin{bmatrix} V_{i-1}^f - V_{i-1}^i \\ V_i^f - V_i^i \\ V_{i+1}^f - V_{i+1}^i \end{bmatrix} \cdot C_G \quad (3.3)$$

$$\lambda = \frac{C_c}{C_g} \quad (3.4)$$

Bei der Betrachtung einer Leitung in einer typischen busähnlichen Struktur, wie einem NoC-Link, sind zwei benachbarte Aggressoren auszumachen. Wenn von einer synchronen digitalen Schaltung ausgegangen wird, sind fünf Muster dieser Kapazitätsveränderungen identifizierbar. Aufgrund der möglichen Schaltkombinationen ergibt sich je MCF eine unterschiedliche Anzahl an Mustern (siehe Tabelle 3.2). Diese führen zu einer Erhöhung (3C, 4C) oder einer Verringerung (0C, 1C) der Verzögerung gegenüber dem nominalen Wert, der bei einem MCF von 2 auftreten würde. Die folgende Veränderung der Verzögerungszeiten auf den Leitungen (Jitter) kann Setup- sowie Holdtime-Fehler zur Folge haben (siehe Abbildung 3.2). Des Weiteren kann es zu Glitches kommen, wenn die Opferleitung nicht schalten soll und von ihren Aggressoren durch deren Signalwechsel Spannungsänderungen ausgelöst werden.

Die Auswirkungen von Crosstalk auf busähnliche Leitungsstrukturen werden in den beiden folgenden Abschnitten genauer untersucht.

### 3.3.1 Messung mittels digitaler Hardware

Die erwähnten Veränderungen der Signallaufzeiten durch Crosstalk-Effekte lassen sich während des Entwurfs der Leitungen durch aufwändige Simulationen oder vereinfach-

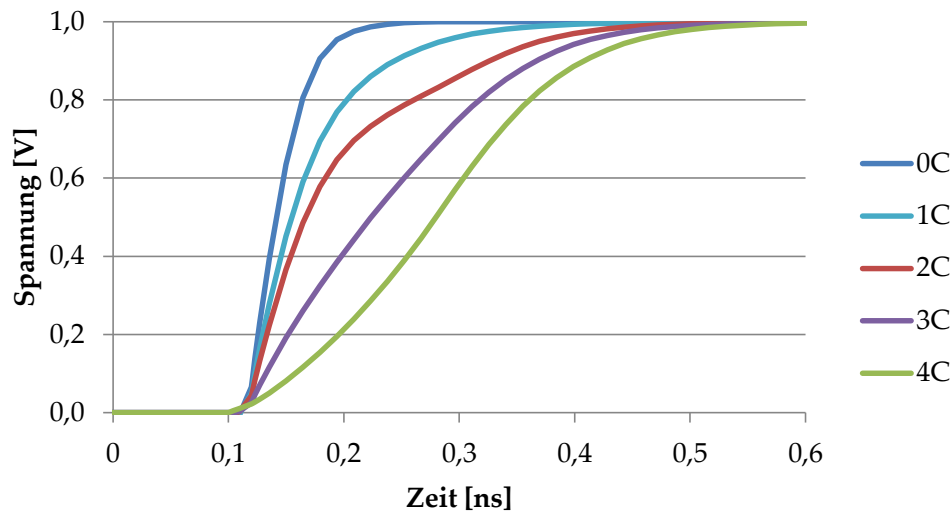


Abbildung 3.2: SPICE-Simulation des Crosstalk-Effekts, Spannungsverlauf der Opferleitung bei verschiedenen MCF

chende Abschätzungen ermitteln. Die Messung der realen Verzögerungen innerhalb des gefertigten Chips kann mit Hilfe von speziellen Messschaltungen erfolgen. Diese müssen allerdings auch die Crosstalk-Muster zum richtigen Zeitpunkt auf den benachbarten Leitungen erzeugen und stellen hohe Anforderungen an die Messgenauigkeit. Es muss mit hoher Abtastrate und genauen A/D-Wandlern gearbeitet werden. Dies ist vor allem für moderne CMOS-Prozesse aufwändig. Allerdings kommt auch eine digitale Messung der Verzögerung durch Crosstalk-Effekte in Frage, z. B. mit Hilfe eines Phasendifferenz-Zählers [SCH<sup>+</sup>00]. Auch die Verwendung eines Ring-Oszillators und eines Frequenzzählers kommen in Frage [TTIM08]. Auf Basis dieser Messmethode wurden Test-Chips produziert, um den Einfluss von Crosstalk sichtbar zu machen.

Die praktischen Auswirkungen von Crosstalk auf das Zeitverhalten der Leitung lassen sich also in gewissem Maße auch mit digitaler Schaltungstechnik sichtbar machen. Dabei sollten auch FPGAs als Untersuchungsobjekt dienen können. Bereits seit dem Jahr 2001 gibt es Arbeiten zu Crosstalk-Effekten auf FPGAs und wie diese gemindert werden könnten [Wil01, RMS03]. Seitdem wurde auf diesem Bereich jedoch nicht intensiv geforscht. Es wird davon ausgegangen, dass die Hersteller ihre Produkte so auslegen, dass keine negativen Auswirkungen zu erwarten sind. Des Weiteren wurden spezielle

FPGA-Zellen zur Erkennung von Crosstalk entwickelt [MPR01]. Die entstandene Erkennung der Symptome von Crosstalk-Effekten funktioniert dadurch jedoch nur für ein bestimmtes FPGA-Modell, das mit der vorgeschlagenen Technik entworfen wurde. Der Crosstalk-Effekt innerhalb von FPGA wurde zudem auch als nützlicher Effekt genutzt, um einen Zufallszahlengenerator zu entwerfen [TCBS09].

Offizielle Informationen von den großen FPGA-Herstellern sind zum Thema Crosstalk nicht verfügbar. Die Dokumentationen und Tools behandeln Crosstalk nur als Off-Chip-Problem zwischen den Pins oder Drähten auf der Leiterplatte [Xil08]. Das physikalische Layout der Leitungen eines kommerziellen FPGA in aktuellen Schaltungstechnologien ist nicht frei verfügbar. Dies verhindert eine simulative Abschätzung der Crosstalk-Verzögerungen. Im Folgenden wird deshalb eine neuartige Messschaltung beschrieben, die in FPGAs implementierbar ist und die Crosstalk-Verzögerung auf den Leitungen bestimmen kann.

Zunächst muss ein Weg gefunden werden die zeitliche Verzögerung eines Signals zu bestimmen. So genannte Tapped Delay Lines eignen sich zur Messung der zeitlichen Verschiebung von einmaligen Ereignissen, indem sich einander entgegenlaufende Signale an den „taps“ abgegriffen und verglichen werden. Die Messgenauigkeit liegt bei Verwendung von aktuellen FPGAs und möglichst guter Kalibrierung zwischen 5 und 50 ps [ZHZ06, JS08]. Je nach Implementierung der Messschaltung könnte diese Genauigkeit ausreichen. Die Implementierung, Kalibrierung und Verifikation einer solchen Schaltung ist jedoch zur Messung eines periodischen Ereignisses sehr aufwändig. Wenn eine reproduzierbare oder wiederkehrende zeitliche Abweichung gemessen werden soll, kann dies auch mit einfacheren Methoden, wie der Bestimmung der Frequenzdrift, stattfinden. Der Crosstalk-Effekt muss dazu in einer mit bestimmter Grundfrequenz schwingenden Schaltung gezielt hervorgerufen werden. Als schwingende Schaltung bietet sich ein Ringoszillator an, der sich sehr einfach aus digitalen Grundfunktionen, die in FPGAs zur Verfügung stehen, aufbauen lässt.

Ein Ringoszillator ist ein Schaltkreis, der aufgrund seiner inhärenten Logikfunktion schwingt. Durch die Verwendung einer ungeraden Anzahl von digitalen Schaltelementen, die eine invertierende Funktion realisieren, kann ein solcher Oszillator implementiert werden. Für ein FPGA bedeutet dies, dass die Look Up Tables (LUTs) als Inverter konfiguriert werden und eine Reihe von LUTs zu einer Inverterkette verschaltet wird. Der Ausgang des letzten Elementes wird auf den Eingang des Ersten zurückgeführt.

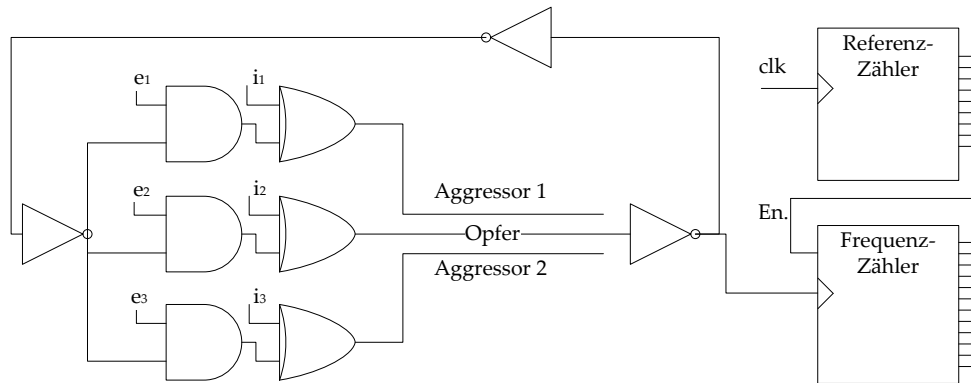


Abbildung 3.3: Messschaltung zur Bestimmung der Verzögerung durch Crosstalk-Effekte

Ein einfacher Weg, um die Frequenz des Oszillators mit rein digitalen Elementen zu messen, ist die Verwendung von zwei Zählern. Einer von ihnen erzeugt eine definierte Zeitspanne mit Hilfe einer bekannten Taktrate. Diese Referenz wird verwendet, um den zweiten Zähler, welcher die positiven Flanken an einer bestimmten Position im Ringoszillator registriert, zu aktivieren (siehe Abbildung 3.3).

Die Frequenz der Schwingung ist von allen vorkommenden Verzögerungen in dem Ring von Elementen abhängig. Wenn durch einen oder mehrere Mechanismen die Verzögerung verändert wird, kann eine Frequenzdrift des Ringoszillators am Ausgang des zweiten Zählers beobachtet werden. Die Einflüsse auf die Verzögerung können unterschiedlicher Art sein. Die meisten von ihnen gehen auf eine Veränderung von der Temperatur und die Variation der Versorgungsspannung zurück. Dies verändert die Schaltgeschwindigkeit der Transistoren. Ein weiterer Grund für eine Änderung der Verzögerung sind externe Einflüsse, wie z. B. benachbarte Schaltelemente und Leitungen, die Crosstalk-Effekte hervorrufen.

Mit der beschriebenen Schaltung lassen sich verschiedene Parameter messen, wenn sichergestellt werden kann, dass die verbleibenden Betriebsbedingungen konstant bleiben. Man kann beispielsweise relative Temperaturänderungen bestimmen, wenn die Betriebsspannung der Schaltung sich während der Messung nicht verändert. So lange hingegen die Temperatur in gewissem Maße konstant bleibt, kann man an der Veränderung der Oszillatorfrequenz eine Änderung der Betriebsspannung ablesen. Diese Art der Messung besitzt verschiedene Fehlerquellen, die die Genauigkeit der Methode beeinträchtigen. Unter Anderem ist auch ein Einfluss von Übersprechen auf Leitungen zwischen den In-

vertoren des Ringoszillators möglich. Dieser Einfluss ist bei einem herkömmlich erstellten FPGA-Design nur sehr gering, da die Wahrscheinlichkeit, dass Leitungen auf langen Strecken benachbart liegen und zum gleichen Zeitpunkt Signalwechsel erfolgen für die meisten Hardware-Entwürfe sehr klein ist.

Um die maximalen Auswirkungen von Crosstalk dennoch sichtbar zu machen, wurde der Ringoszillator um eine Teststrecke erweitert. Die Leitung zwischen zwei Invertern dient als Opfernetz (siehe Abbildung 3.3). Die Aggressoren werden von zwei Leitungen gebildet. Damit die Signale der drei Leitungen möglichst synchron auftreten, wird der Ausgang des Inverters als Eingang für alle drei Leitungen genutzt. Um verschiedene Crosstalk-Muster erzeugen zu können, werden in alle Leitungen XOR und AND-Gatter eingefügt. Damit lassen sich die Aggressoren einzeln aktivieren und die Phase des Signals um  $180^\circ$  verschieben. So können alle Muster durch die Wahl der richtigen Eingänge für die XOR- und AND-Gatter erzeugt werden. In die Opfer-Leitung wurden die Gatter nicht aus funktionalen Gründen eingefügt, sondern damit die Signale möglichst synchron auftreten und nicht durch die Verzögerung der Gatter eine unerwünschte Phasenverschiebung auftritt.

Für die Umsetzung der beiden Messkreise mit Ringoszillatoren wurde VHDL als Beschreibungssprache verwendet. Da die gängigen Tools eine Inverterkette im Zuge verschiedener Optimierungsschritte entfernen würden, müssen die betreffenden Signale im Quelltext entsprechend attribuiert werden. Die Synthese-Software wird auf diese Weise eingeschränkt und sieht vom Entfernen der Signale zwischen den Invertern ab. Für die Umsetzung wurden die Xilinx-Tool-Umgebung und die Entwicklungs-Boards ML507 und ML605 als Zielplattformen genutzt. Sie enthalten Virtex 5 und Virtex 6 FPGAs. Diese Geräte werden in 40 bzw. 65 nm-Technologie durch die Hersteller UMC und Toshiba produziert und sollten einen messbaren Crosstalk-Effekt zeigen.

Um ein Höchstmaß an Kontrolle über Synthese-, Platzierungs- und Routingprozesse zu erlangen, wurden über so genannte „User Constraint Files“ (UCF) Vorgaben gemacht. Dies begrenzt die möglichen Optionen der verwendeten Entwurfs-Tools. Um den Ring-Oszillator an einer bestimmten Position im FPGA zu platzieren, wurden auf diese Weise die Positionen der LUTs der Inverter vorgegeben. Für die Messung der Crosstalk-Effekte war es wichtig die Aggressoren als tatsächliche Nachbarn des Opfernetzes auf einer möglichst langen Strecke zu routen. Für diese Aufgabe musste die Leitungswahl manuell bearbeitet werden.

Die Schaltmatrizen der verwendeten FPGAs sind mit verschiedenen Routing-Kanal-Typen verbunden. Diese Leitungen verbinden verschieden weit entfernte Configurable Logic Blocks (CLB) miteinander. Für Virtex 5 FPGAs werden Double-, Pent-, Long- und Global-Links unterschieden. Sie überbrücken den Abstand von zwei, fünf, 18 und 19 Logikblöcken. FPGAs des Typs Virtex 6 enthalten Double-, Quad-, Long- und Global-Links, welche Blöcke im Abstand von zwei, vier, 16 und 19 CLB verbinden. Um einen Satz von unterschiedlich langen Teststrecken zu erhalten, werden mehrere (1, 2, 3 und 4) der Long-Links kombiniert.

Die Steuerung der Messschaltung erfolgt per Software über ins FPGA integrierte Prozessoren (PowerPC des Virtex 5, Microblaze-Makro auf Virtex 6). Die Register zur Steuerung des Ring-Oszillators und der Teststrecke werden per „Processor Local Bus“ (PLB) für den Speicherzugriff des jeweiligen Prozessors zugänglich gemacht. Durch diese Lösung lässt sich die Kommunikation vollständig auf dem FPGA realisieren und das Endergebnis z. B. per serieller Schnittstelle an einen PC übertragen.

Für die folgenden Ergebnisse wurden jeweils die gemessenen Frequenzen von 10.000 Durchgängen aggregiert und der Durchschnitt gebildet. Der Referenzzähler ließ den Frequenzzähler 200  $\mu$ s laufen. Der gesamte Messvorgang dauert nur wenige Sekunden. Die Verwendung dieser Messmethode liefert sehr konstante Ergebnisse. Dies zeigt die relative Standardabweichung von lediglich 0,03 % für 4 Long-Links und 0,05 % für 1 Long-Link. Bei der Verwendung von 2 und 3 Long-Links liegen die Abweichungen ebenfalls in diesem Bereich.

Die gemessene Frequenz kann eindeutig mit den eingestellten Crosstalk-Mustern korreliert werden. Es gilt anzumerken, dass eine messbare Beziehung nur für die Linktypen Long nachweisbar ist. Alle anderen Verbindungskonfigurationen können nicht durch Crosstalk beeinflusst werden. Dies deutet darauf hin, dass in den gewählten FPGAs die Link-Typen physisch unterschiedlich implementiert sind. Entweder gibt es Unterschiede in der Gestaltung dieser Verbindungen, z. B. größerer Abstand, Abschirmung oder die benachbarten Links in den verfügbaren Schaltplänen sind physisch keine Nachbarn im realen Layout.

Die Messungen zeigen einen großen Einfluss der Temperatur auf die absolute Frequenz des Ringoszillators. Dies wurde aufgrund der generellen Eignung der Schaltung zur Messung der Temperatur erwartet. Um den Fehler aufgrund von Temperaturdrift während der Messung zu minimieren, werden alle Crosstalk-Muster in Folge mit minimalem zeitlichem



Tabelle 3.3: Frequenzen der unterschiedlichen Crosstalk-Muster für verschiedene Leitungslängen in MHz (1 bis 4 mal Long Link)

aggressor length		0C	1C	2C	3C	4C
Virtex 5:	1x	94,3	94,2	93,9	93,3	93,1
	2x	95,0	94,8	94,6	94,0	93,8
	3x	91,7	90,8	90,1	88,7	87,6
	4x	98,4	96,9	95,5	93,4	91,9
Virtex 6:	1x	142,6	141,5	140,2	139,0	137,7
	2x	145,2	143,0	140,6	138,4	136,5
	3x	150,8	147,7	144,2	141,2	138,6
	4x	144,4	141,1	137,4	134,3	131,7

Abstand konfiguriert und die eigentliche Messung erst nach einer Einschwingzeit von 200  $\mu$ s gestartet.

Die Ergebnisse (siehe Tabelle 3.3) zeigen die ermittelten Frequenzen für verschiedene Leitungslängen (1- bis 4-fach Long-Link) und die unterschiedlichen Crosstalk-Muster. Die stark voneinander abweichenden absoluten Frequenzen sind mit Temperaturunterschieden zu erklären, da die Tests verschiedener Leitungslängen zeitlich nicht unmittelbar aufeinander folgten. Die Messungen für die unterschiedlichen Muster wurden jedoch mit minimaler zeitlicher Verzögerung durchgeführt. Aus diesem Grund ist der Vergleich der relativen Verzögerungsänderung sinnvoll. Die Frequenzspanne bei 4-facher Link-Länge für das Virtex 6 FPGA beträgt 131,7 bis 144,4 MHz für das 4C bzw. 0C Crosstalk-Muster. Dies entspricht einer Veränderung der Leitungsverzögerung von 672 ps. Für das Virtex 5 FPGA sind es 91,9 bis 98,4 MHz, was einer Änderung von 719 ps entspricht.

Ein Vergleich der beiden FPGAs zeigt, dass der Einfluss des Crosstalk im Virtex 5 dominanter ist, was durch eine längere Leitungslänge erklärt werden kann. Zum einen ist die architekturbedingte Länge der Links des Virtex 5 größer. Zum anderen wird dieser in 65 nm-Technologie gefertigt, während beim Virtex 6 eine 40 nm-Fertigung angewandt wurde. Die kleineren Strukturen auf dem Chip sorgen so für eine verringerte Verbindungslänge bei gleicher oder geringerer Anzahl überbrückter Logikblöcke.

Da die Temperaturdrift einen großen Einfluss auf die absolute Verzögerung hat, ist es sinnvoll die verschiedenen Messergebnisse auf das Muster 2C normalisiert zu betrachten,

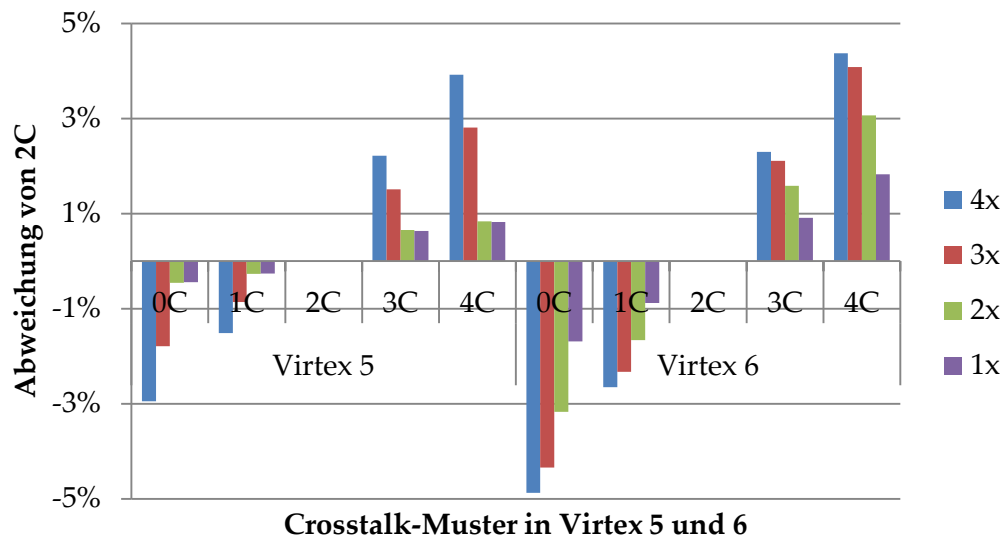


Abbildung 3.4: Auswirkungen der verschiedenen Crosstalk-Muster auf die Oszillator-Frequenzen in Virtex 5 und 6 – für 1, 2, 3 sowie 4-fache Leitungslänge

um die relativen Veränderungen besser vergleichen zu können. Zwei Zusammenhänge werden hier deutlich. Die durch Übersprechen induzierte Verzögerung zeigt eine lineare Abhängigkeit von der Streckenlänge auf der Aggressor- und Opferleitung als Nachbarn verlaufen. Weiterhin hängt die Verzögerung vom Crosstalk-Muster ab. Die 5 Muster zeigen in ihrer bekannten Reihenfolge ebenfalls einen linearen Einfluss, wobei das 0C Muster den größten positiven und 4C den größten negativen Effekt haben.

Der Vergleich der relativen Abweichungen von Virtex 5 und 6 zeigt (siehe Abbildung 3.4) zunächst einen etwas anderen Zusammenhang als die absoluten Zahlen. Der Einfluss auf die absolute Verzögerung im Virtex 5 FPGA ist größer. Da mit dem Virtex 6 jedoch höhere Frequenzen möglich sind, ist die relative Abweichung hier größer. Im Virtex 6 – mit den feineren Fertigungsstrukturen – wird das Übersprechen so eher zu einem kritischen Problem als im Virtex 5. Im Falle des Virtex 5 ist die Verzögerung für 1- und 2-fache Linklänge nahezu gleich. An dieser, von den Erwartungen abweichenden, Stelle scheint eine andere physische Implementierung der Links oder eine fehlerhafte Umsetzung der UCF-Vorschriften vorzuliegen.

Die Validierung der Ergebnisse ist schwierig, da es ohne die Verfügbarkeit des konkreten Layouts der FPGAs keine Möglichkeit gibt, die Ergebnisse abzuschätzen oder die

Messung zu simulieren. Daraus folgt auch, dass keine Angaben über die Genauigkeit, bzw. den systematischen Fehler gemacht werden können. Bei Betrachtung der Verzögerungsänderung unter verschiedenen Crosstalk-Mustern kann aber aufgrund der relativen Werte eine hohe Präzision als ausreichend gelten, um die Ergebnisse als signifikante Bestätigung der Theorie zu bewerten. Die relative Standardabweichung, die in mehreren Messungen auch durch die Temperaturdrift beeinflusst wurde, liegt bei 10.000 Messungen in einem kurzen Zeitraum bei lediglich 0,003 bis 0,005 %. Diese sehr geringe Streuung deutet auf einen niedrigen zufälligen Fehler hin und bestätigt eine ungestörte Messung.

Die Ergebnisse zeigen, dass Übersprechen den Jitter und damit die Signalintegrität in FPGAs stark beeinflussen kann. Weiterhin wurde gezeigt, wie man mit rein digitalen Messmethoden einen grundsätzlich analogen Effekt auf einfache Art und Weise sichtbar machen kann. Mit den beschriebenen Methoden kann man außerdem die Übertragungsstrukturen eines FPGA systematisch testen und eine Verifikation nach der Herstellung durchführen.

#### 3.3.2 Einfluss auf den Leistungsumsatz

Der Leistungsverbrauch einer Leitung hängt von der Schaltwahrscheinlichkeit und den zu ladenden Kapazitäten ab, während die Schaltaktivität von den zu übertragenden Daten bestimmt wird. Die auftretenden Kapazitäten beruhen auf der physikalischen Implementierung und der verwendeten Fertigungstechnologie. Die effektive Kapazität einer Leitung hängt jedoch ebenfalls von den zu übertragenden Datenmustern in Bezug auf die Nachbarn ab (siehe Gleichung 3.1). Daraus lässt sich folgern, dass der durchschnittliche Leistungsverbrauch bei der Übertragung innerhalb eines CMOS-Systems auf zwei Arten von den Datenstatistiken abhängt. Einerseits muss die Schaltwahrscheinlichkeit bekannt sein. Andererseits benötigt man Wissen über die Wahrscheinlichkeit des Auftretens der einzelnen Crosstalk-Muster. Der maximale Leistungsverbrauch hängt von den auftretenden Datenmustern ab, die den höchsten negativen Crosstalk-Einfluss erzeugen.

Die zu beantwortende Frage ist nun, wie viele Informationen über die Daten benötigt werden, um eine für den jeweiligen Zweck genaue Schätzung der Leistungsaufnahme und des Energieverbrauches zu erstellen. Dabei kann als grundsätzliches Ziel der Abschätzung die maximale und die durchschnittliche Leistungsaufnahme bzw. der Energieumsatz unterschieden werden. Weiterhin gilt es herauszufinden, wie genau die Ergebnisse un-

verschiedlicher Abschätzmethoden in Anbetracht der oben genannten Voraussetzungen sind.

Zur Abschätzung der Leistungsaufnahme einer Leitung benötigt man zunächst ein Modell, das die auftretenden Effekte ausreichend detailliert berücksichtigt. Die Kapazitäten von Leitungen werden z. B. vom „Predictive Technology Model“ (PTM) [ZC07] und Frameworks zur Bestimmung des Leistungsverbrauchs [KLPS09] mit Hilfe von geschlossenen Lösungen nach Wong [WLM00] berechnet. Genauere Abschätzungen sind mit aufwändigen FEM-Simulationen möglich und werden für konkrete Technologien von den Herstellern auch zur Verfügung gestellt.

Als NoC-Link-Modell wird wie in den vorangegangenen Kapiteln eine busähnliche Leitungsstruktur angenommen, deren einzelne Leitungen parallel in einer Ebene verlaufen (siehe Abbildung 3.1). Im Folgenden sollen verschiedene Abstraktionsstufen der bekannten Datenstatistiken in Bezug auf die Leistungsaufnahme unter Crosstalk-Einfluss untersucht werden. Um möglichst genaue Werte für die effektiven Koppelkapazitäten und Signalübergänge zu erhalten, ist es notwendig jedes Bit auszuwerten, das den Datenbus durchquert. Außerdem muss eine Analyse der Korrelation mit dem vorherigen Bit dieser Position erfolgen. Da auf der RTL- oder Gatterebene bereits bekannt ist, wie viele Leitungen die Module verbinden, kann die Zuordnung von Signalen zu Leitungen in der busähnlichen Struktur bereits auf diesen Ebenen geschätzt werden. Dabei ist noch kein abschließendes Layout bekannt.

Die Schaltwahrscheinlichkeit der Datenleitungen hängt davon ab, wie zufällig die übertragenen Daten sind. Dabei besitzen komprimierte oder verschlüsselte Daten eine hohe Entropie und verursachen demzufolge mehr Signalwechsel als korrelierte Daten. Als Beispiele für mehr oder weniger komprimierte Anwendungsdaten wurden folgende Datenströme genutzt:

**Worst Case** steht für die Signalstatistik mit der höchsten Leistungsaufnahme. In diesem schlechtesten Fall ist die Schaltwahrscheinlichkeit 100 % und es kommt in jedem Fall das 4C Crosstalk-Muster vor.

**Random** bezeichnet weißes Rauschen, also gleichverteilte Zufallsdaten. Hier ist die Schaltwahrscheinlichkeit 50 %. Die möglichen Crosstalk-Muster treten dadurch ebenfalls zufällig auf.

**JPEG** repräsentiert ein komprimiertes Bild – in diesem Fall ein Landschaftsfoto – nach Joint Photographic Experts Group (JPEG)-Standard. Da die Daten stark komprimiert sind, sollte die Schalthwahrscheinlichkeit nahe 50 % liegen. Darauf, dass bei der verwendeten Kompression bestimmte Crosstalk-Muster häufiger als andere vorkommen würden, gibt es keine Hinweise.

**BMP** ist das gleiche Bild als unkomprimierte Repräsentation mit 24 Bit Farbtiefe. Die Datenstruktur in einer BMP-Datei impliziert, dass sich alle 24 Bit ähnliche Bitmuster wiederholen. Dies gilt, da die Farbwerte (RGB) sich in einem Foto nicht mit jedem Pixel abrupt ändern. Die Schalthwahrscheinlichkeit bei einer Ausrichtung an dieser Wortbreite sollte deshalb deutlich unter 50 % liegen.

**H.264** ist der momentan verbreitetste Video-Codec. Als Quelle diente der Kinotrailer eines Comic-Films. Da eine sehr starke Kompression stattfindet, werden die Ergebnisse vermutlich denen der Random-Daten gleichen und nahe 50 % liegen.

**Video** enthält die gleiche Sequenz als uncodiertes Video. Die Repräsentation entspricht im Prinzip einer Abfolge von BMP-Bildern. Daher ist ein ähnliches Ergebnis zu erwarten. Da ein Comic verwendet wurde, in dem typischerweise einfarbige Flächen vorherrschen, sollte die Schalthwahrscheinlichkeit bei 24 Bit-Ausrichtung noch niedriger als die des Fotos im BMP-Format ausfallen.

**Web Surfing** steht für den Mitschnitt der Netzwerkdaten beim Aufruf einiger Webseiten und repräsentiert somit Datenaufkommen, die z.B. in mobilen Webanwendungen anfallen. Typischerweise ist ein Teil davon unkomprimiert (Header) und der größte Teil komprimiert oder verschlüsselt. Die Schalthwahrscheinlichkeit sollte demnach leicht unter 50 % betragen.

**Text** bezeichnet einen Datenstrom der aus einer Quelltextdatei mit 4.600 Zeilen C-Code besteht. Die Verwendung der ASCII-Codierung lässt eine Korrelation bei allen Wortbreiten mit Vielfachen von 8 Bit erwarten. Da beim normalen Alphabet in der ASCII-Codierung nur 7 Bit verwendet werden, ist außerdem eine um ein Achtel verminderte dynamische Leistungsaufnahme zu erwarten, da jedes achte Bit keine Schalthwahrscheinlichkeit besitzt.

**MP3** bezeichnet einen komprimierten Musiktitel. Ähnlich wie bei JPEG ist hier ein Datenstrom zu erwarten, der zufälligen Daten gleicht.

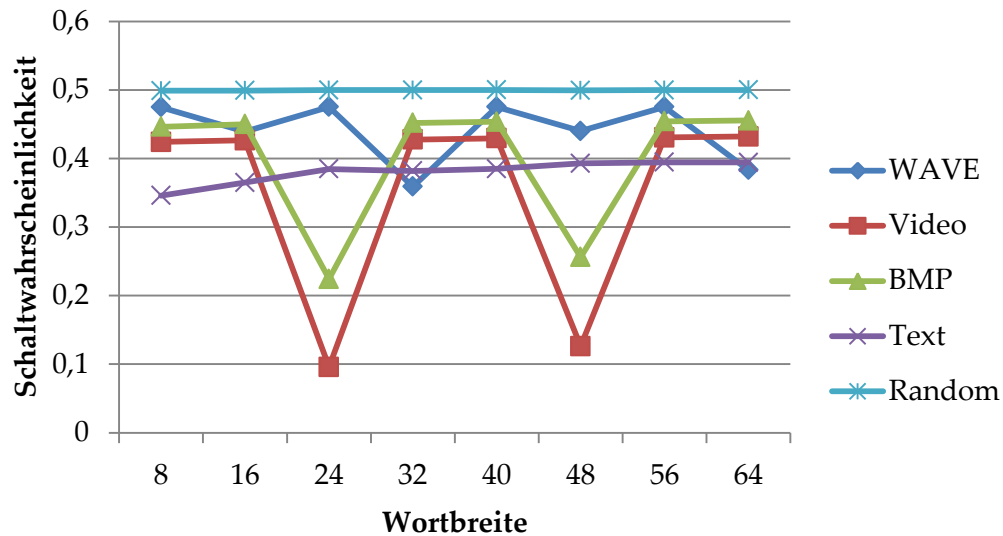


Abbildung 3.5: Schaltwahrscheinlichkeit unterschiedlicher unkomprimierter Daten bei verschiedenen Wortbreiten

**WAVE** beinhaltet den gleichen Musiktitel als unkomprimierte, Pulscode modellierte (PCM) Darstellung. Dabei wird Stereo – also zwei Kanäle mit jeweils 16 Bit verwendet. Dies impliziert eine erhöhte Korrelation, wenn der Datenstrom in Datenwörter der Breite 16 oder 32 Bit zerlegt wird.

**H.264 Decoder SoC** ist der Mitschnitt der Daten eines in mehrere Funktionsblöcke aufgeteilten H.264 Decoders [FLD<sup>+</sup>08]. Die einzelnen Funktionsblöcke und damit auch deren Kommunikation untereinander sind charakteristisch für ein SoC.

Zunächst wurden die unkomprimierten Daten auf ihre Schaltwahrscheinlichkeit untersucht. Wie bereits beschrieben, ist eine starke Abhängigkeit von der gewählten Busbreite zu erwarten. Die Wortbreite wurde schrittweise von 1 bis 8 Byte bzw. 8 bis 64 Bit variiert (siehe Abbildung 3.5). Wie erwartet, erzeugt Random konstante Schaltaktivitäten von 50 % und ist von der gewählten Busbreite unabhängig.

Die Schaltaktivität der Text-Datei liegt ungefähr ein Fünftel unter der der Random-Daten. Mit der Zunahme der angenommenen Bitbreite von 8 bis zu 64 Bit nimmt auch die Schaltwahrscheinlichkeit leicht zu. Diese steigt, da bei Wortbreiten von mehreren Byte eine Häufung von gleichen aufeinanderfolgenden Buchstaben weniger Einfluss hat. Da

ein Quelltext der Programmiersprache C benutzt wurde, gibt es viele Einrückungen. Diese Häufung von aufeinanderfolgenden gleichen Zeichen sorgt für die verringerte Schaltwahrscheinlichkeit. Da in der ASCII-Codierung ein Bit ungenutzt ist, liegt die Schaltaktivität für diese Bitposition bei 0. Eine weitere Absenkung über dieses Achtel hinaus wird durch die Einschränkung auf druckbare Zeichen und den Schwerpunkt auf Kleinbuchstaben und Klammern hervorgerufen. Im Ergebnis bleibt die Schaltwahrscheinlichkeit bei der Übertragung des ASCII-codierten Texts unter 40 % und liegt bei byteweiser Übertragung bei ca. 35 %.

BMP und Video zeigen jeweils ähnliche Abhängigkeiten der Schaltaktivität von der gewählten Wortbreite. Wie erwartet zeigen sich bei Vielfachen von 24 Bit die höchsten Korrelationen, da hier die gleichen Farbwerte auf die jeweiligen Leitungen abgebildet werden. Insgesamt liegt die Aktivität im BMP etwas höher als im unkomprimierten Video, was vor Allem daran liegt, dass als Quelldaten einerseits ein Foto andererseits ein Comic-Ausschnitt verwendet wurde. Im Foto gibt es mehr Rauschen und Farbverläufe, während der Comic vor allem von einfarbigen Flächen dominiert wird. Die Schaltwahrscheinlichkeit sinkt dadurch auf 22 % respektive auf unter 10 %.

Die unkomprimierten Ton-Daten zeigen hohe Korrelationen und damit niedrige Schaltaktivitäten bei Vielfachen von 32 Bit. Hier liegt nur eine Wahrscheinlichkeit von 35 % vor. Etwas geringere aber noch messbare Korrelationen sind auch bei Vielfachen von 16 Bit nachweisbar. Hier werden abwechselnd die Werte des linken und rechten Kanals auf die jeweiligen Leitungen abgebildet. Da sich diese Informationen nur gering unterscheiden, liegt die Schaltwahrscheinlichkeit in diesem Fall bei ca. 44 %.

Zusammenfassend lässt sich festhalten, dass die Schaltwahrscheinlichkeit von der Busbreite und deren Korrelation mit der Datenstruktur der unkomprimierten Daten abhängt. Komprimierte oder zufällige Daten haben eine Schaltwahrscheinlichkeit von ca. 50 %. Die benötigte Energie zur Übertragung von Daten hängt also stark von diesen Eigenschaften ab.

Zum Vergleich der Leistungsaufnahme und der benötigten Energie für die verschiedenen Daten und den Einfluss von Crosstalk-Effekten wird eine Berechnung nach Formel 3.3 für jeden Fall durchgeführt. Zum einen wird die Leistung nur unter Berücksichtigung der Schaltrate bestimmt. Normalerweise gehen konservative Abschätzungen ohne Kenntnis der Datenstatistiken von 50 % Schaltwahrscheinlichkeit aus. Zum anderen wird die Änderung der effektiven Koppelkapazitäten unter Einfluss von Crosstalk berücksichtigt. Nach

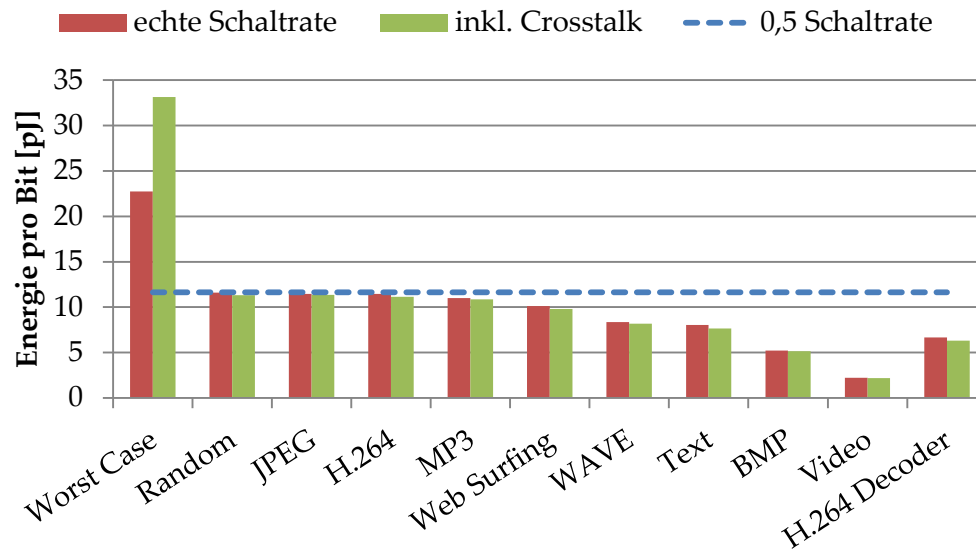


Abbildung 3.6: Durchschnittliche Energie für die Übertragung eines Bits über 200  $\mu\text{m}$  (PTM, einfacher Abstand, mittlere Metalllage) in 65 nm-Prozesstechnologie – hier für einen Bus mit 24 Bit Wortbreite für unterschiedliche Daten und drei verschiedene Methodiken

diesen drei Modellen wird für die beschriebenen Daten eine Abschätzung des Energiebedarfs für eine Übertragung über einen 24 Bit breiten Bus der Länge 200  $\mu\text{m}$  in einem 65 nm-Prozess erstellt.

Die Auswertung (siehe Abbildung 3.6) zeigt eine deutliche Korrelation mit den zuvor dargestellten Schaltwahrscheinlichkeiten bzw. dem Kompressionsgrad der Daten. Hier wird deutlich, dass die Schaltwahrscheinlichkeit große Abweichungen von der 50 %-Methode hervorrufen kann. In diesem Fall – vor allem Video, BMP und Text – liegen die Leistungsaufnahme und damit auch die verbrauchte Energie pro Bit zum Teil weit unter den konservativen Abschätzungen. Die Berücksichtigung von Crosstalk führt zu einem erneuten Anstieg der dynamischen Leistungsaufnahme im Worst Case. Der zusätzliche Anstieg hängt im Wesentlichen von der Größe der Koppelkapazitäten zwischen den Leitungen zu deren Kapazitäten zu statischen Elementen ober- und unterhalb der Leitungsebene ab.

Die Auswertung zeigt, dass im schlechtesten Fall – verglichen mit der 50 %-Abschätzung – mehr als die doppelte Menge an Energie benötigt wird. Diesen Worst Case be-



zeichnet man auch als „Power Virus“ [GJB<sup>+</sup>10, NVK<sup>+</sup>07]. Dieser Fall kann in der Realität vorkommen, sollte jedoch unter normalen Betriebszuständen nicht erreicht werden. Ein Vergleich der Modelle zeigt, dass die Einbeziehung von Crosstalk zu generell etwas niedrigeren Ergebnissen im Bereich der Energie führt. Dies liegt jedoch nicht an der Verteilung der auftretenden Crosstalk-Muster sondern am Busmodell, dass in der höheren Detailstufe die Randkapazitäten der äußeren Leitungen als etwas geringer als die zwischen den Leitungen annimmt, was im Modell ohne Crosstalk nicht der Fall ist. Bis auf diesen Unterschied bringt das Crosstalk-Modell keine Vorteile in der Energieabschätzung. Da sich die positiven und negativen Muster auslöschen, bestimmt im Durchschnitt das 2C-Muster die Energieaufnahme.

Für den durchschnittlichen Leistungsumsatz sind die Werte zu denen der vorangegangenen Analyse der Energie analog. Die maximale Leistungsaufnahme entspricht jedoch eher den Verhältnissen der Worst Case-Untersuchung. Falls das 4C-Crosstalkmuster auf mehreren Leitungen des Busses auftritt und die Schaltwahrscheinlichkeit in Richtung 100 % steigt, wächst die Leistungsaufnahme enorm an. Wenn also die maximale Leistungsaufnahme abgeschätzt werden soll, dann muss der Einfluss von Crosstalk und unnatürlich hoher Schaltaktivität berücksichtigt werden.

#### 3.3.3 Folgen und Gegenmaßnahmen

Crosstalk verstärkt die Datenabhängigkeit des Leistungs- und Energiebedarfs während der Signalübertragung. Außerdem wird durch die Schwankungen der effektiven Kapazität die Übertragungszeit beeinflusst. Je nach Crosstalk-Muster beschleunigt oder verlangsamt sich das Signal. Dies führt dazu, dass bei synchroner Übertragung der Worst Case angenommen werden muss und dies die erreichbare Taktlatenz bestimmt.

Die Maßnahmen zur Verminderung der negativen Auswirkungen finden hauptsächlich auf technologischer Ebene und während des Schaltungslayouts statt. So wird z. B. versucht für die Zwischenräume zwischen den Leitungen ein Material mit möglichst niedriger Permittivität zu nutzen. Wenn genügend Platz vorhanden ist, können während des Layouts auch die Zwischenräume der Leitungen vergrößert oder so genannte Shielding Wires eingefügt werden. Diese Leitungen werden fest mit Masse verbunden und bilden so einen konstanten Nachbarn für die eigentlichen Signalleitungen, auf welchen so kein Crosstalk-Muster mehr erzeugt werden kann. Eine weitere Möglichkeit den Crosstalk-

Effekt abzumildern bildet das gezielte abwechselnde Einfügen von Treibern oder das abschnittsweise Vertauschen der Leitungen [WH10].

Aber auch auf höheren Ebenen kann bereits eine Berücksichtigung der Crosstalk-Problematik erfolgen. Der Einsatz von bestimmten Leitungscodes – Crosstalk Avoidance Codes (CAC) – kann das Auftreten problematischer Datenmuster unterbinden. Es wird beispielsweise der Forbidden Transition Code [PZGG06] angewandt, der entgegengesetzte Signalwechsel auf benachbarten Leitungen innerhalb des Codewortes verhindert. Diese Variante schließt alle Datenmuster aus, die einen MCF von 3 oder 4 erzeugen. Da diese Technik bei großen Bitbreiten zu langen Codewörtern und komplexen De- und Encodern führt, kann der Datenbus auch in kleinere Gruppen partitioniert werden. Auf diese Weise können jeweils 3 Bit zu 4 Bit codiert werden. Dies führt jedoch dazu, dass an den Gruppengrenzen wiederum Maßnahmen gegen auftretende Crosstalk-Muster getroffen werden müssen, wovon die einfachste das Duplizieren der Randleitung ist. Schließlich erhält man bei 32 Bit-Links 53 codierte Leitungen [PZGG06]. Eine weitere Möglichkeit bietet das Ausschließen von Mustern die im Falle eines Übergangs zu bestimmten Crosstalk-Übergängen führen würden. Diese Forbidden Pattern sind beispielsweise '010' und '101'. Damit kann man 4 Bit zu 5 Bit codieren und erhält für einen 32 Bit-Bus 52 Leitungen [PZGG06]. Außerdem können beispielsweise Fibonacci-Reihen genutzt werden um einen CAC zu generieren. Mit dieser Technik werden weitaus weniger zusätzliche Leitungen benötigt. Aus einem 32 Bit-Bus würde ein 47 Bit-Code erzeugt. Der benötigte Hardware-Aufwand ist jedoch so hoch, dass sich dieser Code praktisch nicht sinnvoll einsetzen lässt [Was12, DLK10]. Des Weiteren kann die Codierung neben der Eigenschaft Crosstalk zu unterdrücken, weitere Funktionen erfüllen – z. B. die Fehlererkennung und -korrektur [GPB09]. Die genannten Codes verhindern das Auftreten des MCF 4 bzw. 3 und 4. Damit wird eine geringere Signallaufzeit und Leistungsaufnahme im Worst Case sowie ein geringerer Energiebedarf im Mittel erreicht. Allerdings entsteht ein Mehraufwand durch die steigende Anzahl an Leitungen und die nötigen De- und Encoder. Dies führt dazu, dass eine Abwägung, ob sich ein CAC lohnt, je nach Leitungslänge, Busbreite, Flächenkosten und fertigungstechnisch bedingter Quantität des Crosstalk-Effektes getroffen werden muss.

Aus den erlangten Erkenntnissen der Crosstalk-Messung innerhalb von FPGAs lässt sich schließen, dass in bestimmten Fällen auch in programmierbaren Schaltungen unerwünschte Crosstalk-Phänomene auftreten und die Signalübertragung empfindlich stören

können. Wie bereits erwähnt, lassen sich in den offiziellen Dokumentationen der FPGA-Tools keine Hinweise darauf finden, dass Crosstalk berücksichtigt wird. Da der Einfluss der effektiven Kapazitäten in aktuellen Technologien jedoch sehr hoch sein kann, sollte dies während des Routing-Vorgangs zumindest auf den kritischen Pfaden des Entwurfs beachtet werden. Dieses Vorgehen wurde bereits vorgeschlagen [DRR11] und Timing-Verbesserungen von bis zu 3,5 % erzielt [Wil01]. Sollte der Routing-Algorithmus Crosstalk vermeiden, könnte die hier beschriebene Messmethode genutzt werden, um die Effizienz dieser Gegenmaßnahme physisch zu überprüfen.

### 3.4 Fazit

Die durchgeführten Untersuchungen haben gezeigt, durch welche Mechanismen in passiven Verbindungsstrukturen Leistungsumsatz und Energieaufnahme erzeugt werden. Da die Leitungen eines NoC-Links ähnlich wie in Bussen parallel zueinander verlaufen, können Busmodelle verwendet werden, um diese zu modellieren. Die Leistungsaufnahme der Leitungen wird durch das Auf- und Entladen der gebildeten Kapazitäten gebildet. Bei bestimmten Anforderungen an Leitungslänge oder Signalqualität müssen Treiber eingefügt werden, die zusätzlich Leistungsumsatz erzeugen. Die gesamte Leistungsaufnahme des Links lässt sich demnach durch das Ermitteln der Leistungskapazitäten und der Treiberdimensionierung ermitteln. Außerdem muss die Schaltaktivität bekannt sein. Eine Schaltwahrscheinlichkeit von 50 % stellt sich als gute obere Abschätzung heraus, wobei festzustellen ist, dass oft eine weitaus geringere Aktivität anzutreffen ist.

Weiterhin wurde betrachtet, welche Auswirkungen die gegenseitige Beeinflussung der Leitungen durch Crosstalk haben kann. Eine eigens entwickelte Messschaltung zeigte die Folgen des Phänomens in FPGAs. Im Bereich der Leistungsaufnahme konnte gezeigt werden, dass die Datenmuster, die Crosstalk auslösen, in zu erwartendem Übertragungsverkehr nicht gehäuft vorkommen. Der negative und positive Einfluss des Effektes hebt sich somit im Mittel auf. Die durchschnittliche Leistungsaufnahme wird nicht durch Crosstalk beeinflusst. Es konnte jedoch auch gezeigt werden, dass die maximale Leistungsaufnahme bei Verwendung eines Worst Case-Datenmusters sehr stark ansteigen kann.

Um den Leistungsumsatz, der durch einen Link zwischen den Routern eines NoC entsteht, zu bestimmen, werden die Kapazitäten der Leitungsstrukturen und die Schalt-

statistiken der zu übertragenden Daten benötigt. Wenn die Abschätzung bereits auf Systemebene erfolgen soll, müssen erste Annahmen zum Layout bzw. zu den Ausmaßen des späteren Chips und der verwendeten Technologie vorliegen. Daraus können die Länge der Leitungen und eventuell nötige Treiberdimensionierungen gewonnen werden. Die Schaltaktivitäten sind in diesem Stadium oft noch nicht genau bekannt. Aus diesem Grund eignet sich für die Leistungsaufnahme über einen größeren Zeitraum die 50 %-Abschätzung als obere Grenze der Betrachtungen.

# Kapitel 4

## NoC-Router

In diesem Kapitel wird der zweite grundlegende Bestandteil eines NoC genauer untersucht. Zunächst wird auf die Grundlagen eines NoC-Routers eingegangen. Dazu gehört die Funktionalität, mit der die Wegewahl der Pakete und die Auswahl der Ausgangsports festgelegt wird. Außerdem werden weitere Aufgaben, wie die Pufferung von Paketdaten und die gezielte Bevorzugung bestimmter Pakete, betrachtet.

Anschließend wird eine konkrete Implementierung eines NoC-Routers genauer untersucht. Zunächst wird die Implementierung und ihre Parameter genauer beschrieben und eine Optimierung für niedrigen Leistungsumsatz vorgenommen. Schließlich wird die Synthese mit dem Tool „Design Compiler“ von SYNOPSYS genutzt, um den Router-Entwurf nach verschiedenen Aspekten zu beurteilen. Als Zieltechnologie werden drei verschiedene Fertigungsgenerationen genutzt. Dabei handelt es sich um „CMOS065“, „CMOS045“ und „CMOS032“ der 65, 45 respektive 32 nm-Generationen von STMicroelectronics, die unter Verwendung der jeweils zugehörigen Standardzellbibliotheken genutzt werden.

Schließlich wird am Ende dieses Kapitels ein Konzept zur Verwendung der auf Zellebene erlangten Erkenntnisse auf Systemebene entwickelt und mit anderen existierenden Lösungen verglichen.

Teile der in diesem Kapitel verwendeten Ergebnisse wurden ebenfalls in der eigenen Publikation [GWG<sup>+</sup>13] genutzt.

## 4.1 Grundlagen

Die Knotenpunkte eines NoC werden durch Router gebildet. Ihre Hauptaufgabe liegt in der Sicherstellung der Wegewahl, die das Datenpaket zum gewünschten Empfänger weiterleitet. Die Regeln, nach denen sich der entstehende Pfad ergibt, werden als Routing-Verfahren oder -Algorithmen bezeichnet. Sollte die Entscheidung der Wegewahl nicht in den Knotenpunkten sondern z. B. von einer zentralen Instanz getroffen werden, bezeichnet man sie als Switches, da sie nur eine Weiterleitung auf einen vorgegebenen Port realisieren. Um eine Weiterleitung des Paketes entlang mehrerer Router zu erreichen, müssen diese untereinander kommunizieren. Dabei kann es sich z. B. um die Mitteilung handeln, dass die Daten eines neuen Paketes auf der Leitung anliegen oder darum, ob die angelegten Daten erfolgreich übernommen wurden. Diese Funktion des Routers bezeichnet man als Flusskontrolle. Je nach Umsetzung des Routers kann dieser weitere Funktionen wie die Pufferung von Paketeilen umsetzen.

In welcher Form die Router untereinander mit Links verbunden sind bestimmt die Topologie des Netzes. Im Falle eines NoC wird vor allem zwischen unregelmäßigen und regelmäßigen Topologien unterschieden. Welche hauptsächlich von heterogenen respektive homogenen NoCs genutzt werden. Gängige NoC-Topologien unterscheiden sich im Grad der Vernetzung. Würfel sind demnach beispielsweise mehr vernetzt als Torus- oder Gitternetz-Formen. Da sich eine zweidimensionale Topologie am einfachsten auf einen herkömmlichen Chip abbilden lässt, wird bisher meist das 2D-Gitternetz gewählt [SKH08]. Mit dem Aufkommen von 3D-Techniken wie dem Aufeinanderstapeln von „Dies“ werden auch 3D-Topologien attraktiver. Auch in wissenschaftlichen Betrachtungen sowie kommerziellen Produkten ist bisher vor allem die 2D-Mesh-Topologie verwendet worden (siehe Kapitel 2.5). Aus diesen Gründen beziehen sich die folgenden Betrachtungen vor allem auf diese Struktur.

Die implementierte Routingmethode muss eine Dead- und Livelockfreiheit sicherstellen. Das bedeutet, dass die Pakete im Netz sich nicht blockieren bzw. unendlich viele Hops zum Ziel benötigen können dürfen. Dies ist bei einfachen Methoden, die das Turnmodell [GN92] erfüllen, der Fall. Mit dem Turn-Modell werden Routingvorgänge in einem Mesh-Netzwerk als Folge von acht möglichen Richtungswechseln beschrieben. Damit eine Dead- und Livelockfreiheit erreicht wird, sind nur solche Turns erlaubt, die sich nicht zu einem Zyklus kombinieren lassen. Das einfachste Beispiel ist das XY-Routing, bei dem nur vier mögliche Richtungswechsel erlaubt sind. Weitere Möglichkeiten bieten teilweise

Tabelle 4.1: Entscheidungstabelle bei XY-Routing

Adresse X	Adresse Y	Routing-Entscheidung
> als aktuelle X	–	Ost
< als aktuelle X	–	West
= aktuelle X	> als aktuelle Y	Nord
= aktuelle X	< als aktuelle Y	Süd
= aktuelle X	= aktuelle Y	IP-Core

adaptive Verfahren mit einer größeren Auswahl an Turns, wie West-First, North-Last oder Negative-First. Bei diesen sind jeweils sechs Richtungswechsel erlaubt. Durch die zusätzlichen Auswahlmöglichkeiten ergeben sich mehr mögliche Pfade. Damit kann ein sich an Ausfälle oder Überlastsituationen anpassendes Routingverfahren implementiert werden.

Eine weitere Möglichkeit Livelocks zu verhindern, ist die Einführung eines Zählers für die zurückgelegten Hops. Eine Verfallsgrenze beschränkt die maximale Lebenszeit des Pakets. Danach wird das Paket verworfen. Bei Verwendung dieser Methode muss allerdings ein Kontroll-Mechanismus in der Ende-zu-Ende-Kommunikation implementiert werden, damit sichergestellt ist, dass die Pakete auch ankommen bzw. der Sender vom Fehlschlag der Sendung in Kenntnis gesetzt wird. Deadlocks können auf ähnliche Weise ausgeschlossen werden. Diese Methoden werden vor allem bei nicht minimalen Routingmethoden wie Flooding eingesetzt.

Das XY-Routing ist das am weitesten verbreitete Routing-Verfahren in NoCs mit Mesh-Topologie [SKH08]. Im Paketheader ist die Zieladresse als Bit-Folge gespeichert. Diese Adresse lässt sich in einen Teil für die Ziel-Zeile und einen für die Ziel-Spalte zerlegen. In jedem Knoten wird nun überprüft, ob die Zielspalte größer oder kleiner als die aktuelle Router-Adresse ist (siehe Tabelle 4.1). Je nach dem Ergebnis dieses Vergleichs wird nach Osten oder Westen weitergeleitet. Sollten die X-Adressen identisch sein, ist die richtige Spalte erreicht und es wird die Zielzeile mit der aktuellen Router-Adresse verglichen. Hier wird nun auf gleiche Weise verfahren. Ist auch die richtige Zeile erreicht wird zum Port der Ressource weitergeleitet.

Nachdem die Routingentscheidung getroffen wurde, müssen die Daten vom Eingangs- auf den Ausgangs-Port weitergeleitet werden. In welcher Weise dies geschieht, wird

durch die Switching-Methode bestimmt. Generell sind zwei Verfahren zu unterscheiden. Das Circuit-Switching sorgt in einer Initialisierungsphase dafür, dass ein Pfad reserviert wird. Dieser kann dann garantiert genutzt werden. Der Vorteil dieser Methode ist, dass prinzipiell kein Pufferspeicher nötig ist und eine geringe Latenz auftritt. Der zusätzliche Aufwand durch den Aufbau des Pfades macht es jedoch nur für lange Pakete attraktiv. Die zweite Methode wird als Packet-Switching bezeichnet. Hier wird der Port erst bei Ankunft des Pakets ausgewählt. Es können weitere Unterverfahren unterschieden werden. Wenn der erste Teil des Paketes den Port reserviert und der Weg durch den Router bis zum letzten Teil des Paketes garantiert geschaltet bleibt, spricht man von Wormhole-Switching. Der Vorteil ist, dass nur kleine Pufferspeicher notwendig sind und eine geringe Latenz auftritt. Es kann jedoch zu Stauungen über den gesamten Weg kommen, was die Netzwerkauslastung unnötig negativ beeinflussen kann. Das Store-and-Forward-Switching speichert vor der Weiterleitung erst das ganze Paket ab. Dies hat den Nachteil, dass je nach maximal erlaubter Paketgröße enorme Mengen an Speicher in jedem einzelnen Router zur Verfügung stehen müssen. Bei Nutzung dieses Verfahrens tritt eine große Erhöhung der Latenz auf. Virtual-Cut-Through-Switching bildet eine Kombination aus beidem, es stellt genügend Speicher für ein ganzes Paket zur Verfügung, sollte jedoch im nächsten Knoten genug Kapazität zur Verfügung stehen, wird sofort weitergeleitet. Verwendet wird meist Wormhole-Switching [SKH08], da die Implementierungskosten hier am niedrigsten sind.

Die bereits angesprochene Flusskontrolle sorgt für die Kommunikation zwischen den Routern. Meist wird ein Handshake-Verfahren wie Request/Acknowledgement (Req/Ack) verwendet [SKH08].

Auf einer höheren Netzwerkebene kann ebenso eine Flusskontrolle in der Ende-zu-Ende-Kommunikation implementiert sein. Hier können ebenfalls Verfahren wie Req/Ack, Fehlerkorrektur oder Checkpointing, wie sie aus anderen Netzwerkprotokollen bekannt sind, umgesetzt werden.

An der Eingangs- oder der Ausgangs-Seite des Routers erfolgt normalerweise eine Pufferung der Daten. Je nach Routing, Switching, und Flusskontrolle sind hier verschiedene Anforderungen an die Kapazitäten des Zwischenspeichers gefragt. Für Store-and-Forward-Switching muss etwa ein ganzes Paket der maximalen Länge abgespeichert werden. Für das meist verwendete Verfahren Wormhole-Switching ist es nur nötig einen Teil des Paketes (Flow Control Unit, Flit) abzuspeichern. Bestimmte Verfahren kommen prinzipiell



auch ohne Pufferspeicher aus [MM09]. Aus Performance-Gründen wird jedoch ab einem bestimmten Punkt eine durch Pipelining implizierte Pufferung eingeführt. Zumeist wird der nötige FIFO-Speicher je nach Größe als Register oder SRAM implementiert.

Die Anschlüsse der Links an den Router bezeichnet man als Ports. Es wird zwischen Ein- und Ausgangsports unterschieden. Wie viele Ports ein Router besitzt, hängt stark von der Topologie des NoC ab. Bei den meist verwendeten 2D-Mesh-Netzen haben die Router typischerweise fünf Ein- und Ausgangsports. Jeweils einen für jede Himmelsrichtung und einen zum Anschluss eines IP-Core.

Virtuelle Links oder Kanäle (virtual channels, VC) dienen zur effizienteren Ausnutzung der physischen Links. Eine Vervielfachung der Puffer im Router führt dazu, dass mehrere Pakete gleichzeitig – und deren Flits unter Umständen abwechselnd – über einen physischen Kanal transportiert werden können. Dies ist ohne VC bei Verwendung von Wormhole Switching nicht möglich. Dieses Vorgehen ermöglicht eine höhere Auslastung der Links bzw. eine unterschiedliche Priorisierung der Daten – Quality of Service (QoS). Die Kosten dieser Lösung sind beträchtlich. Der ohnehin schon pufferdominierte Router vergrößert seine Fläche um ein Vielfaches, falls nicht komplexere Speicherstrukturen implementiert werden, die mehrere Pufferstufen unter den VC aufteilen können [YCPC10]. Die zu erreichenden Frequenzen im NoC sinken ebenfalls, da zusätzliche Logik zum Multiplexen der Flits nötig ist. Ein weiterer Nachteil ist das Durchmischen der Flits unterschiedlicher Pakete auf den Links. Dadurch wird eine wahrscheinliche Korrelation in einem Paket mit unkomprimierten Daten (siehe Abschnitt 3.3.2) durch das Einstreuen von Flits anderer Pakete durchbrochen. Oftmals wird aus diesen Gründen für die Einführung von QoS anstelle VC die Verwendung von mehreren parallelen NoCs [CGRB06, GCTK13] vorgeschlagen. Auch in bereits umgesetzten Manycore-Systemen werden meist mehrere NoCs genutzt [Til13, Ada13], um beispielsweise Daten intern, zum externen Speicher oder Informationen zum Systemmanagement auf unterschiedliche Weise zu transportieren.

## 4.2 Verlustleistungsschätzung

Die Leistungsaufnahme eines NoC-Routers kann in verschiedenen Phasen des Entwurfsablaufes ermittelt werden. Typischerweise steigt die Genauigkeit der Abschätzungen mit der fortschreitenden Entwicklung von der algorithmischen Ebene über die Hardwarebeschreibung bis hin zu den konkreten Layout-Daten an. Während der frühen Entwurfsphasen

wird die Funktion des NoC algorithmisch beschrieben. Eine qualitative Aussage zum Leistungsbedarf ist hier durchaus schon möglich. Zumindest die relationale Ordnung verschiedener Leistungskategorien eines NoC kann bestimmt werden.

Quantitative Aussagen sind ohne eine gewisse Rückannotation – also das Bestimmen von einigen Leistungswerten in unteren Ebenen – nicht in befriedigendem Maße möglich. Da zu diesem Zeitpunkt der Kommunikationsvorgang meist als abstraktes Transaktionsmodell umgesetzt wird, gibt es noch keine Informationen über Paketformate oder gar Bit-Statistiken. Aus diesen Gründen bekommen simulative gegenüber analytischen Verfahren auf den unteren Entwurfsebenen mehr Gewicht. Hier besteht die Möglichkeit Signalstatistiken direkt mit der entsprechenden Leistungsaufnahme zu erfassen.

Wenn möglich sollte eine Kombination der Verfahren genutzt werden, die die Vorteile der verschiedenen Ebenen vereinen. Hier wird deshalb darauf zurückgegriffen, die NoC-Router auf niedrigen Ebenen zu charakterisieren und eine Rückannotation auf höhere Ebenen durchzuführen. Dies kombiniert eine hohe Genauigkeit mit der Geschwindigkeit von Simulationen auf Systemebene. Konkret sollen verschiedene NoC-Router bis auf Gatternetzebene synthetisiert werden. Danach soll eine simulative Bestimmung der Leistungsaufnahme in bestimmten Situationen erfolgen. Dies betrifft vor allem die Auslastung des Routers.

### 4.3 Verwendetes Router-Design

Der in allen folgenden Untersuchungen verwendete NoC-Router und das darauf aufbauende NoC realisiert jeweils fünf Vollduplex-Verbindungen (10 Links) mit zur Entwurfszeit einstellbarer Bitbreite. Der Router selbst arbeitet mit einer einfachen xy-Routing-Logik und einem Pufferspeicher an jedem Eingangsport. Es sind fünf Ports vorgesehen, womit sich ein skalierbares 2D-Mesh-Netzwerk realisieren lässt, bei dem jeder Router eine Ressource (R) und vier andere Router, bezeichnet nach den Himmelsrichtungen (N, O, S, W), anbindet. Eine Erweiterung durch den Anschluss von zusätzlichen Ressourcen an die Router am Rand ist ohne Änderung der Router-Architektur möglich [Kub11]. Die Flusskontrolle zwischen den Routern wird über zwei Signale implementiert. Das REQ-Signal zeigt an, dass ein Flit am Port anliegt. Ein ACK-Signal in die entgegengesetzte Richtung signalisiert, dass die Daten übernommen werden konnten. Dieser Zwei-Wege-Handshake und damit

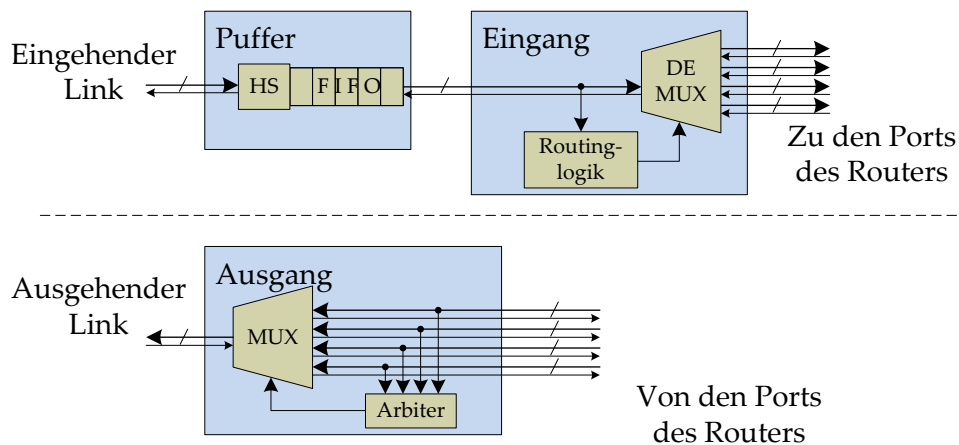


Abbildung 4.1: Bestandteile eines Ports des NoC-Routers

auch die Datenübermittlung zwischen den Routern dauert zwei Taktzyklen. Sollte auf den Links ein Pipelining nötig sein, vergrößert sich die Latenz entsprechend.

Der Router selbst lässt sich in die drei einzelnen Module Puffer, Eingangslogik und Ausgangslogik aufteilen (siehe Abbildung 4.1). Diese gibt es jeweils für jeden Port. Der Pufferspeicher arbeitet als First-in-First-out-Speicher (FIFO). Er wird durch Register implementiert, deren Tiefe zur Entwurfszeit einstellbar ist. Die Eingangslogik trifft die Routingentscheidung auf Basis der Zieladresse und der eigenen Adresse. Danach steht fest auf welchen Ausgangsport geschaltet werden muss. Ein entsprechender Demultiplexer realisiert diese Funktion. Die Ausgangslogik arbitriert einen Eingangsport mit reihum wechselnder Priorität (Round-Robin-Methode), falls mehrere Pakete gleichzeitig auf dem jeweiligen Ausgangsport eintreffen. Ein Multiplexer schaltet die Eingänge der verschiedenen Ports auf den Ausgangsport. Das Herzstück des Routers, die Verschaltung der unterschiedlichen Ports – die so genannte Crossbar – ist durch die Multiplexer ausgeführt.

Innerhalb des Routers gibt es kein Pipelining, dies wäre aber möglich, falls die Routing-Logik ein zu großes Delay verursachen sollte. Die Latenz der Datenübertragung beträgt so nur einen Taktzyklus. Der Timing-Pfad geht dabei vom Eingangspuffer über die Routing-Logik und den Arbitrier und dann durch das Steuersignal zurück in die Eingangs-Ports und Puffer, also einmal vollständig durch den Router.

Der maximal mögliche Datendurchsatz des Routers lässt sich mit der wählbaren Link-Breite einstellen. Einem Auftreten von sich zeitweise gegenseitig blockierenden Paketen bei Überlastsituationen kann mit der Erhöhung der Puffertiefe des FIFO-Speichers be-

Tabelle 4.2: Übersicht der wählbaren Designparameter

Parameter	Werte
Link-Breite	32, 64, 128 Bit
Puffertiefe je Port	1, 2, 4, 8
Routing-Algorithmus	XY
Adressbreite	4, 6, 8 Bit
mögliche NoC-Größe	2x2 bis 16x16
Arbitrierung	Round-Robin
Switching	Wormhole
Flusskontrolle	REQ/ACK Handshake
Betriebsfrequenzen	100 bis 1000 MHz
Fertigungstechnologie	32, 45 und 65 nm

gegnert werden. Alle gewählten und einstellbaren Entwurfsparemeter sind in Tabelle 4.2 zusammengefasst.

#### 4.3.1 Verbesserungen des Router-Designs

Der vorgestellte Entwurf eines NoC-Routers ist bewusst einfach gehalten. Er unterstützt nur die wichtigsten Funktionen, um mit geringem Platzbedarf und einfacher Architektur ein geeignetes Untersuchungsobjekt für die folgenden Betrachtungen zu bieten. Trotzdem sorgen die folgenden einfachen Optimierungsschritte für ein weitaus effizienteres Design.

Da Pakete nach dem XY-Routing zunächst in X-, dann in Y-Richtung weitergeleitet werden, kann ein in Y-Richtung laufendes Paket nicht mehr die Richtung wechseln. Von den acht möglichen Richtungswechseln im 2D-Mesh sind somit nur 4 möglich, wenn man den Ressourcen-Port nicht betrachtet. Das führt dazu, dass die Crossbar der Router entsprechend vereinfacht werden kann. Die Eingangs-Ports von Nord und Süd benötigen keine Verbindungen nach Ost und West. Dadurch fällt der Multiplexer an diesen Stellen jeweils nur halb so groß aus. Die Routing-Logik in diesen Ports kann ebenfalls reduziert werden. Folglich fällt in den Ausgangsports von Ost und West der Multiplexer und die Arbitrier-Logik entsprechend kleiner aus. Da der Routing-Algorithmus diese Optimierungen bereits impliziert, können EDA-Tools eventuell die Reduzierung automatisch vornehmen. Dies war jedoch in Tests mit Xilinx- und Synopsys-Werkzeugen nicht vollständig der Fall.

Der NoC-Router benötigt Pufferspeicher, der in diesem Fall in Form von Registern ausgeführt ist. Der Betrieb dieser Flipflops kann den Energieumsatz dominieren. Auch wenn keine neuen Daten durch die Ports fließen, wird die Schaltaktivität durch das Taktsignal in den Puffern für Leistungsaufnahme sorgen. Eine Möglichkeit dies einzuschränken ist, den Takt der Register für die nicht benötigte Zeit abzuschalten. Das dafür benötigte Signal kann aus dem Handshake-Signal erzeugt werden. Dieses zeigt ohnehin an, wenn neue Flits anliegen oder alte Daten aus dem Puffer benötigt werden.

Das generierte Signal wurde zunächst mit den Enable-Eingängen der Register verbunden. Dies zeigte jedoch nicht den gewünschten Effekt, da die interne Schaltung der Register den aktiven Takt weiterhin auf einige Transistoren durchschaltet. Die Leistungsaufnahme ist deshalb auch bei deaktiviertem Enable und ohne Änderung der Dateneingänge sehr hoch. Die erwartete, deutlich verringerte Standby-Leistungsaufnahme konnte nur mit einem Clock-Gating erzeugt werden. Das dazu benötigte Signal, welches den Takteingang der Register nur aktiviert, wenn ein neues Flit abgespeichert oder von der Routing-Logik ausgelesen werden muss, wird mit dem Taktsignal verknüpft. Dies bedeutet einen geringen zusätzlichen Flächenbedarf. Da zusätzliche Gatter eingefügt werden, die potentiell den kritischen Pfad verlängern, ist eine leichte Reduktion der maximal erreichbaren Frequenz zu erwarten.

Eine darüberhinausgehende Verbesserung durch Clock-Gating könnte erzielt werden, wenn man den Füllstand des FIFO-Puffers berücksichtigt und den Takteingang der einzelnen Stufen separat aktiviert. Weiterhin enthält die Routing- und Arbitrier-Logik einige Register um den Zustand des jeweiligen Automaten für Wormhole-Switching und Round Robin-Arbitrierung zu speichern. Diese könnten ebenfalls vom Takt getrennt werden, um die Leistungsaufnahme im Ruhezustand noch weiter abzusenken.

In den folgenden Abschnitten werden vier verschiedene Implementierungen verglichen, die sich durch die unterschiedlichen Optimierungen unterscheiden (siehe Tabelle 4.3).

## 4.4 Performance

Der eingangs beschriebene NoC-Router wird bezüglich seiner Leistungsfähigkeit zunächst anhand der erreichbaren Frequenz analysiert. Die Performance des sich ergebenden Gesamtsystems aus mehreren Routern [Cor09] soll hier nicht Gegenstand sein.

Tabelle 4.3: Verwendete Implementierungen des NoC-Routers

Abkürzung	Clock-Gating	reduzierte Ports
NG-NR	∅	∅
NG-R	∅	✓
CG-NR	✓	∅
CG-R	✓	✓

Die erreichbaren Frequenzen einer exemplarischen Router-Konfiguration wurden für die Synthese für ein FPGA – Xilinx Virtex 6 – und für Standardzellbibliotheken in 32, 45 und 65 nm ermittelt. Dabei wird – wie in allen folgenden Untersuchungen in diesem Kapitel – nur der Router als einzelnes Modul getestet. Eventuelle Eingangs- oder Ausgangsbausteine werden nicht einbezogen, da der Router innerhalb des NoCs ebenfalls als Modul eingefügt wird und nicht als einzelner Chip vorliegen wird. Die erzielten Werte hängen von den Optionen und Zielsetzungen des jeweiligen Synthesewerkzeugs ab. Auch haben kleine Änderungen des Hardwareentwurfs teils unerwartete Auswirkungen auf die Optimierungsergebnisse der Syntheselgorithmen. Die erreichbare Frequenz, die hier zur Bewertung der Performance herangezogen werden soll, wird durch den kritischen Pfad bestimmt. Dieser liegt zwischen den Registern der Eingangspuffer bis hin zum Ausgangs-Port und passiert dabei Routing-Logik und Arbiter. Der sich anschließende Link wird nicht berücksichtigt.

Im Falle der FPGA-Synthese wird die Erwartung bestätigt, dass die für XY-Routing optimierten Varianten mit reduzierter Crossbar (R) schneller realisiert werden können als die normalen Konfigurationen (siehe Abbildung 4.2). Entgegen der Erwartung meldet die Zeitanalyse des Routers mit Clock-Gating (CG) eine höhere mögliche Taktfrequenz. So ergibt sich die insgesamt beste Konfiguration mit Clock-Gating und reduzierter Crossbar (NG-R). Dieses Ergebnis ist jedoch auf einen Fehler innerhalb des Tools für die Taktanalyse zurückzuführen. Die einzelnen abschaltbaren Taktsignale innerhalb des Routers werden vom Synthese-Tool unabhängig voneinander betrachtet, obwohl sie zwingend synchron zu behandeln wären. Das Taktsignal kann zwar durch ein zusätzliches Gating-Register geleitet werden und damit eventuell besser an die folgenden Speicherelemente verteilt werden kann, dies dürfte die erreichbare Taktfrequenz an dieser Stelle jedoch nicht steigern.

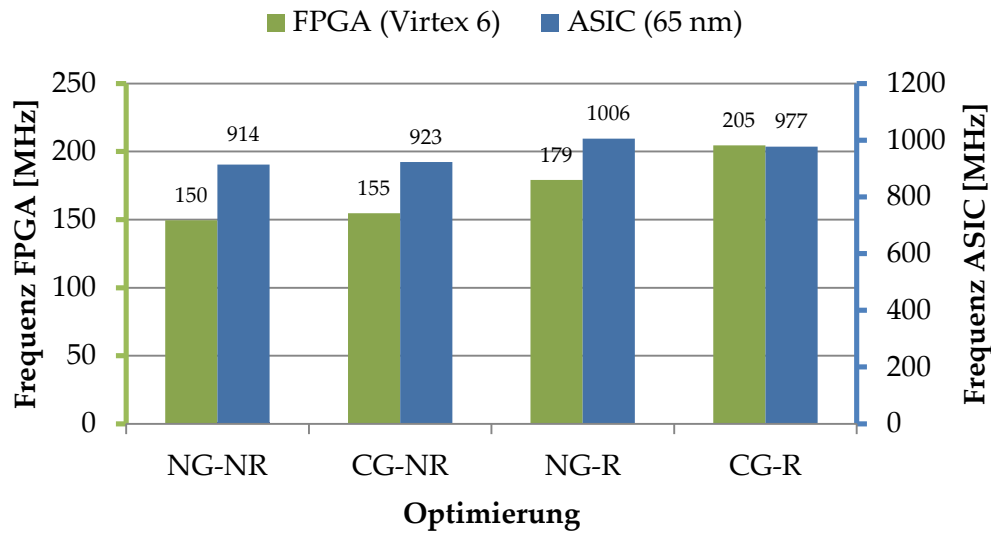


Abbildung 4.2: Performance des Routers mit Zielplattform FPGA und ASIC (64 Bit, Puffertiefe: 2)

Die reduzierte Crossbar (R) erzielt im Falle der beispielhaften ASIC-Synthese für 65 nm-Standardzellen ebenfalls einen Geschwindigkeitsvorteil. Das Clock-Gating bringt jedoch nicht in jedem Fall eine Verbesserung der erzielbaren Frequenz mit sich. Die Aufteilung des Takt-Signals durch das Clock-Gating scheint in der FPGA-Technologie besser implementierbar zu sein. Weiterhin haben die Syntheseoptionen und Heuristiken der Optimierungstools hier einen anderen bzw. größeren Einfluss als im Falle des FPGA.

Das schnellste Design in ASIC-Standardzellen ist in diesem Beispiel NG-R. Festhalten lässt sich jedoch, dass die Portreduzierung einen größeren Geschwindigkeitsgewinn bringt als das Clock-Gating, welches eher auf eine optimierte Leistungsaufnahme zielt (siehe Kapitel 4.6). Die Zielfrequenzen bis zu 1 GHz konnten in der 32 nm-Technologie unter Verwendung aller Parameterkombinationen (siehe Tabelle 4.2) erreicht werden. Die Zellbibliotheken mit 45 und 65 nm konnten bei Anwendung der einfachen Optimierungsoptionen nicht in allen Fällen 1 GHz erreichen. Dies betrifft vor allem große Bitbreiten und hohe Puffertiefen, die den kritischen Pfad verlängern.

## 4.5 Fläche

Der Flächenbedarf des Routers ist ein entscheidender Faktor für die Kostenabschätzung, Effizienz und Realisierbarkeit des Kommunikationssystems. Vor allem im Falle eines größeren NoC kann die Fläche der Summe aller Router ein entscheidendes Distinktionsmerkmal gegenüber anderen Konfigurationen oder alternativen Kommunikationslösungen darstellen. Weiterhin kann der Flächenbedarf einen ersten Anhaltspunkt für die zu erwartende Leistungsaufnahme geben. Ausnahmen werden durch stark schwankende Schaltaktivitäten gebildet, wie sie bei der Verwendung von Clock-Gating auftreten können.

Die Synthese-Ergebnisse für FPGA sind wiederum getrennt von denen der Standardzellbibliotheken zu betrachten. Das verwendete Xilinx FPGA ist aus einer Vielzahl von LUT und Registern aufgebaut, die mit Schaltmatrizen verbunden sind. Je nachdem wie gut sich bestimmte Logikfunktionen auf ein und dasselbe LUT-Register-Paar abbilden lassen, kann der Flächenverbrauch unterschiedlich ausfallen. Im Folgenden wird deshalb die Anzahl der verwendeten Register und LUT getrennt angegeben. Weiterhin gilt es zu beachten, dass die Ergebnisse nur für die verwendete Zielplattform gelten. Aufgrund der Eigenheiten der verschiedenen FPGA-Architekturen, z. B. unterschiedlich viele Eingänge eines LUT, lassen diese sich untereinander nicht ohne weiteres vergleichen.

Anhand der Synthese-Ergebnisse für ein Xilinx Virtex 6 FPGA bestätigen sich die erwarteten Verbesserungen (siehe Abbildung 4.3). Die Reduzierung der Ports eines Routers mit 64 Bit-Links und einem zweistufigen Puffer auf die XY-spezifische Hardware ändert den Bedarf an LUT von 2842 auf 1930, was einer Einsparung von 32 % entspricht. Die Anzahl der Register verringert sich um 4 bzw. 0,5 %.

Der Einfluss auf die Anzahl der Register ist geringer, da der größte Teil dieser auf die Pufferspeicher am Eingangsport entfällt. Die Puffer werden durch die Port-Reduktion jedoch nicht beeinflusst. Die Routinglogik und der Round-Robin-Arbitrer besitzen nur eine sehr geringe Zahl an Registern, von denen allerdings noch einige wenige eingespart werden konnten.

Die Reduzierung der Ports hat für die ASIC-Implementierung ebenfalls die größte Verbesserung zur Folge (siehe Abbildung 4.4). Das Weglassen einiger Multiplexer und Register reduziert die Anzahl der benötigten Zellen je nach Puffertiefe um 10 bis 33 %. Die Puffertiefe hat auf diese Ersparnis kaum Auswirkungen. Die Bitbreite der Links hingegen hat einen nahezu linearen Einfluss auf die Größenänderung.



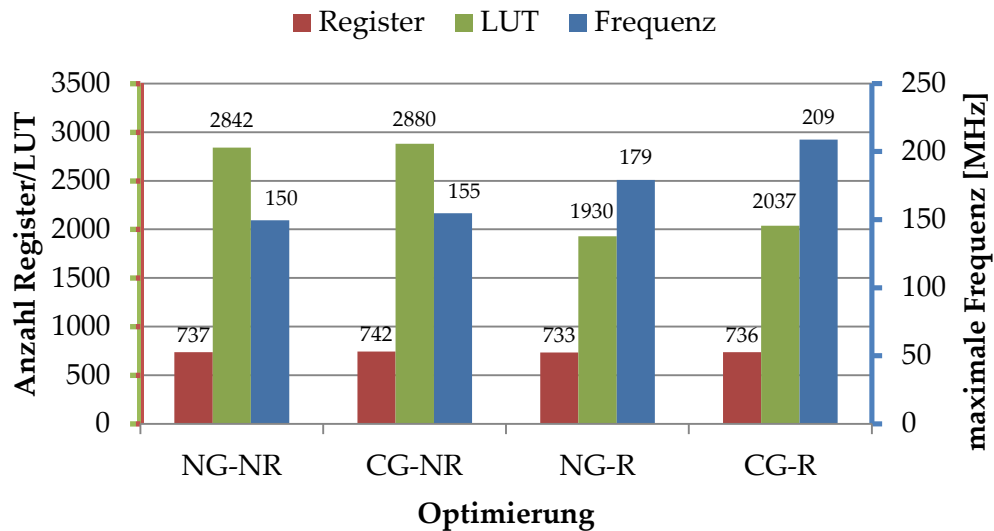


Abbildung 4.3: Flächenbedarf und Frequenz des Routers für FPGA

Absolut belegen die Router-Varianten in 32 nm-Technologie eine Fläche zwischen 2100 und 38 000  $\mu\text{m}^2$ . Bei einer quadratischen Chipgröße mit einer Kantenlänge von 10 mm ergibt sich so für z. B. ein 10x10 NoC ein Router-Flächenanteil von 0,2 bis 3,8 %. Obwohl für das gesamte NoC zusätzlich Fläche durch Links und deren Treiber belegt wird, ist der Anteil sehr gering und der Router kann damit durchaus als flächeneffizient bezeichnet werden.

Das Clock-Gating bringt einen Mehrbedarf von 15 bis 470 Zellen mit sich. Dabei ist der Unterschied von der Anzahl der Pufferstufen und der Bitbreite abhängig. Die Erhöhung dieser Parameter fügt zusätzliche Flipflop-Zellen ein, die mit dem generierten Taktsignal versorgt werden müssen. Je mehr Pufferspeicher dies betrifft, desto mehr zusätzliche Gatter müssen eingefügt werden, da die Treiberstärke des Clock-Gating-Gatters nicht ausreicht. Zusammenfassend lässt sich festhalten, dass die Implementierung ohne Clock-Gating mit reduzierten Ports (NG-R) den geringsten Flächenbedarf aufweist. Mit Clock-Gating ist der Flächenbedarf geringfügig größer, die erhofften Verbesserungen im dynamischen Leistungsumsatz sollten dies jedoch rechtfertigen.

Die Implementierung einer Testkonfiguration (32 Bit, Puffertiefe 1) in verschiedenen ASIC-Technologien zeigt, dass die Anzahl der benötigten Zellen nahezu konstant bleibt

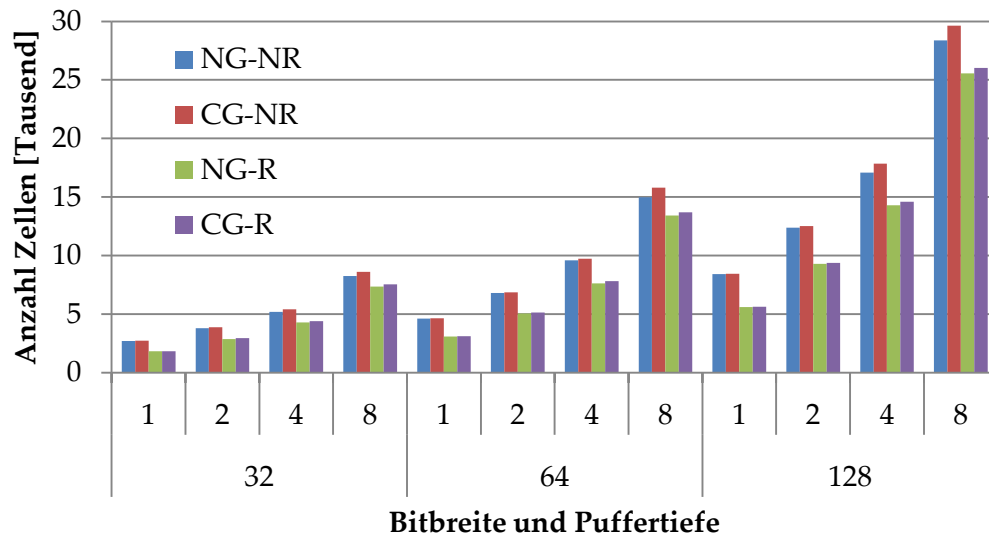


Abbildung 4.4: Flächenbedarf der verschiedenen Router-Varianten am Beispiel der 32 nm-Technologie

(siehe Abbildung 4.5). Was eine logische Konsequenz daraus ist, dass die Auswahl an Zellen in den Bibliotheken weitgehend identisch ist. Der Umfang der Bibliotheken unterscheidet sich lediglich in einigen Komplexgattern, was in kleineren Unterschieden bei der Gesamtzahl der verwendeten Zellen resultiert. Die Gesamtfläche der entstandenen Router sinkt wie zu erwarten mit Verwendung neuerer Technologien. Einerseits sind die Strukturgrößen und damit die einzelnen Zellen kleiner. Andererseits müssen in älteren, größeren Technologien stärkere und damit größere Gatter bzw. Treiberzellen verwendet werden, da die angestrebten Schaltgeschwindigkeiten sonst nicht erreicht werden.

## 4.6 Leistungsaufnahme

Die Leistungsaufnahme des NoC-Routers ist ein entscheidender Faktor für seine Einsetzbarkeit, denn der Anteil der Kommunikation bestimmt zunehmend das gesamte Leistungsbudget eines Many-Core-Prozessors. Im Folgenden werden zunächst die dynamische Leistungsaufnahme des vorgestellten NoC-Routers und danach der durch Leckströme erzeugte statische Leistungsumsatz betrachtet.

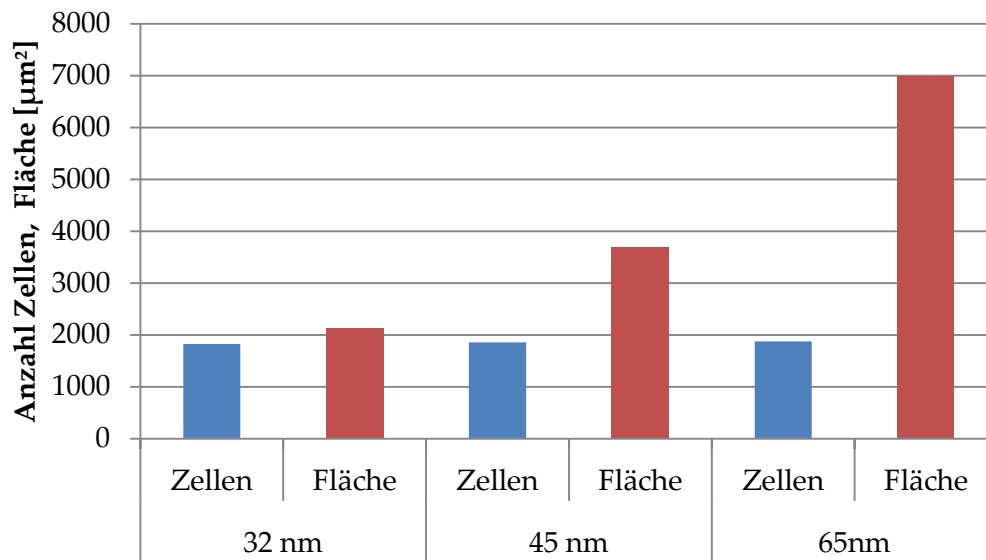


Abbildung 4.5: Flächenbedarf der unterschiedlichen ASIC-Technologien (32, 45, 65 nm)

Der entworfene Router wird aus Leistungssicht von seinen Pufferspeichern dominiert. Die Folge ist, dass sich der Leistungsumsatz überwiegend aus dem Schalten der Register zusammensetzt. Die Analyse des Routers ohne Clock-Gating zeigt diese Auswirkungen deutlich (siehe Abbildung 4.6). Im Beispiel (Konfiguration: 65 nm, 128 Bit, Puffertiefe 4, 500 MHz) zeigt sich der aktivitätsabhängige dynamische Leistungsbedarf. Getestet wurden die Aktivitäten 0, 1 und 2 aktive Ports. Dies entspricht der Weiterleitung von keinem, einem bzw. zwei Flits gleichzeitig. Während die Ein- und Ausgangslogik jeweils nur 10 bis 600 µW benötigen, sind es beim Puffer 12 bis 17,2 mW. Im Ruhezustand, wenn kein Paket weitergeleitet wird, findet auch keine Schaltaktivität auf den Datenleitungen statt. Dieser Leistungsbedarf resultiert nur aus dem Taktnetzwerk und den internen Schaltvorgängen an den Zellen mit Takteingang (Register). Da es in Ein- und Ausgangslogik nur sehr wenige Register gibt, sticht somit der Pufferspeicher, der hier in Form von Registern realisiert ist, stark hervor. Prozentual macht der Puffer etwa 95 bis 98 % des Leistungsbedarfs aus.

Als sinnvolle Betriebskonfiguration muss deshalb von Clock-Gating für die Puffer ausgegangen werden. Dies sorgt dafür, dass der Leistungsbedarf im Ruhezustand auf 2 % absinkt. Der Anteil des Pufferspeichers macht nun nicht mehr 98 sondern nur noch 31 % und damit ähnlich viel wie der der Ein- und Ausgangsmodule aus. Wenn ein oder zwei

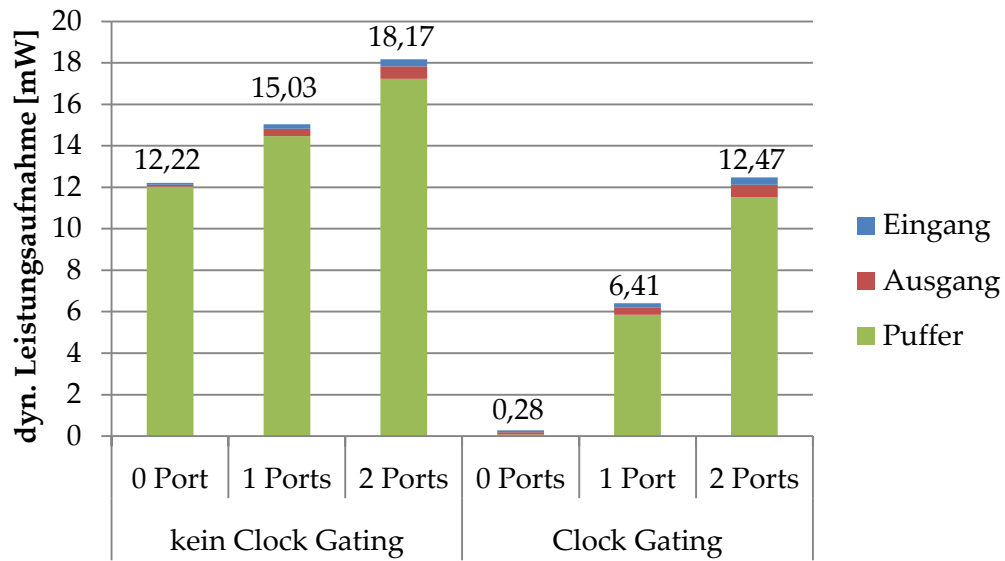


Abbildung 4.6: Aufschlüsselung der dynamischen Leistungsaufnahme des NoC-Routers mit und ohne Clock-Gating, bei 0, 1 und 2 aktiven Ports

Ports gleichzeitig aktiviert sind – was zwei gleichzeitig zu verarbeitenden Flits entspricht – können immer noch 57 % respektive 31 % eingespart werden. In diesen Fällen werden die Puffer der betroffenen Ports mit dem Taktsignal versorgt und machen wiederum einen großen Teil der Leistungsaufnahme aus (91 bzw. 92 %). Die Auslastung eines Routers kann je nach Verkehrssituation sehr unterschiedlich sein. Beim hier implementierten XY-Routing wurde jedoch nach eingehender simulativer Untersuchung festgestellt, dass die Anzahl der gleichzeitig vom Router zu verarbeitenden Pakete selten mehr als zwei beträgt. Bei vorliegender Sättigung zeigte sich, dass im Worst-Case die Wahrscheinlichkeit für das Auftreten von mehr als einem Flit zur gleichen Zeit bei unter 20 % liegt [GGWT11]. Für mehr als 2 Pakete wird deshalb im Folgenden (siehe Abschnitt 4.6.1) eine Extrapolation vorgeschlagen.

Über die verschiedenen Parameter erreicht das Clock-Gating eine Reduktion der Leistungsaufnahme auf ca. 1/5 (kleinster Router) bis 1/100 (größter Router) im Ruhezustand. Dies gilt für alle drei getesteten Technologien.

Die Leistungsaufnahme steigt prinzipiell mit der benötigten Fläche an, da mehr Gatter genutzt werden. Die dynamische Leistung hängt jedoch stark von der Schaltaktivität ab. Daher ist ein Unterschied zwischen Router-Implementierungen mit und ohne Clock-

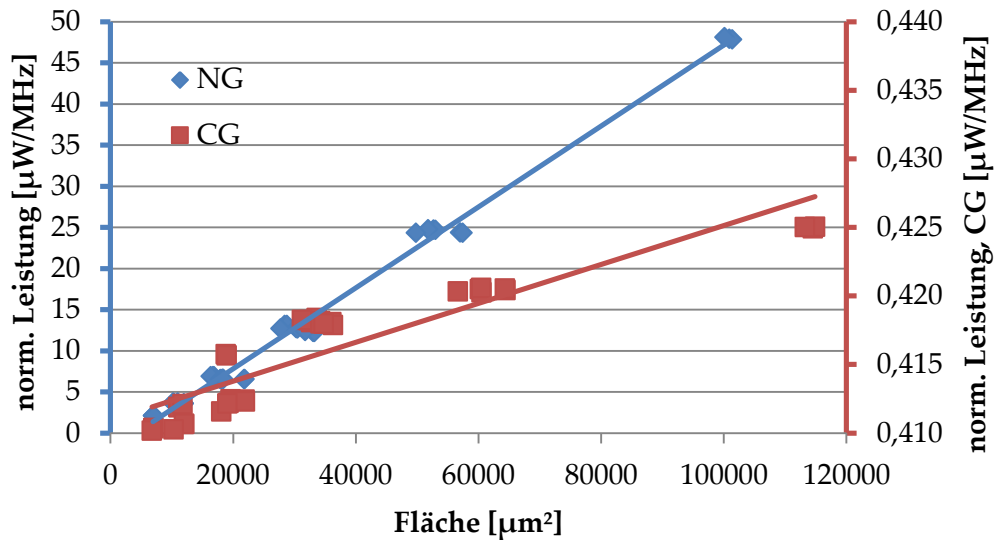


Abbildung 4.7: Zusammenhang von Flächenbedarf und dynamischer Leistungsaufnahme des Routers mit und ohne Clock Gating

Gating zu erwarten. In Abbildung 4.7 ist die dynamische Leistungsaufnahme auf die Taktfrequenz normiert und über der benötigten Fläche dargestellt. Es werden Implementierungen mit und ohne Clock-Gating ohne aktive Ports (Ruhezustand) verglichen.

Die Leistungsaufnahme steigt bei Implementierungen ohne Clock-Gating mit der Fläche ungefähr linear an. Obwohl der Leistungsumsatz der Router-Variante mit Clock-Gating um ein Vielfaches geringer ist, fällt ein Unterschied im Vergleich mit der Regressionsgeraden im Diagramm auf. Ein großer Teil der Fläche, also die Register der Pufferspeicher, wird hier nicht mehr mit dem Taktsignal versorgt. Dies führt dazu, dass der dynamische Leistungsbedarf nicht mehr ohne weiteres mit der Fläche korreliert.

Aufgrund des verwendeten Switching-Mechanismus gibt es zwei verschiedene Arten von Flits. Die Header-Flits leiten bei Ankunft im Router die Pfadwahl und damit die Port-Arbitrierung ein. Dabei wird in den entsprechenden Logik-Modulen Leistung umgesetzt. Die darauf folgenden Body-Flits werden auf den festgelegten Pfad weitergeleitet ohne Schaltaktivität in der Routing- und Arbitrierlogik auszulösen. Der ermittelte Unterschied der Leistungsaufnahme zeigt einen Unterschied je nach Größe der Routerimplementierung von 20 bis 30 %. Dieser Unterschied macht sich je nach Paketlänge auch bei der Betrachtung

auf Paketebene bemerkbar. Kurze Pakete weisen demnach eine hohe Abweichung auf, während Pakete mit vielen Body-Flits eine geringere Abweichung zur Folge haben.

Während der Optimierungen des Routerdesigns (siehe Kapitel 4.3.1) wurden Änderungen an der Portkomplexität eingeführt. Einige Ports haben durch diese Optimierungen auf XY-Routing kleinere Zustandsspeicher in Arbiter und Routinglogik. Diese Änderungen führen dazu, dass die Leistungsaufnahme während der Durchleitung eines Flits davon abhängt von welchem auf welchen Port die Daten geleitet werden. Im Diagramm (siehe Abbildung 4.8) sieht man die Leistungsaufnahme für alle Kombinationen von Ein- und Ausgangsport des Routers, sortiert nach Eingangsport. Die Werte differieren in diesem Beispiel (32 Bit, Clock-Gating, 1 Port aktiv, Puffertiefe: 2) um maximal  $5,7 \mu\text{W}$ , der maximale Unterschied zwischen den für jeden Eingangsport gemittelten Werten ist  $4,5 \mu\text{W}$ . Prozentual entspricht der Unterschied jedoch nur 2,2 bzw. 1,7 %.

In Abbildung 4.9 sind die Werte nach Ausgangsport sortiert. Hier stellt man neben der maximalen Differenz von  $5,7 \mu\text{W}$  eine durchschnittliche Abweichung nach Ausgangsport von  $1,1 \mu\text{W}$  fest. Prozentual entspricht dies 2,2 bzw. 0,4 %.

Es lässt sich eine eindeutige Korrelation zwischen Leistungsaufnahme und Eingangsport feststellen. Eine Möglichkeit die Genauigkeit der Leistungsvorhersage zu beeinflussen, bietet sich darin, die einzelnen Ports, die an der Transaktion beteiligt sind, bzw. nur die Eingangsports zu berücksichtigen. Falls man diesen Zusammenhang nicht berücksichtigt und einen Durchschnittswert über alle Ports annimmt, entsteht folglich ein maximaler Fehler von ca. 1,1 %. Lässt man den Eingangsport in die Leistungsschätzung einfließen, beträgt der Fehler ca. 0,5 % ( $1,4 \mu\text{W}$ ). Die Steigerung der Präzision beträgt also 0,6 Prozentpunkte. Da der entstehende Fehler bei Annahme eines Durchschnittswertes für alle Ports jedoch nur 1,1 % beträgt, gilt es abzuwägen, ob diese Genauigkeitssteigerung einen effektiven Aussagegewinn für die Gesamtabschätzung auf Systemebene bringt oder nicht. Außerdem werden höhere Anforderungen an die Simulationsumgebung gestellt, sollten diese Details mitberücksichtigt werden.

Die statische Leistungsaufnahme der Router-Hardware ist von der Anzahl der verwendeten Gatter – also der Fläche – abhängig. Daneben spielen Temperatur, Betriebsspannung und Schwellspannung bzw. Fertigungstechnologie eine Rolle (siehe Kapitel 2.2). Wenn der Router Pakete verarbeitet, ist der Anteil des Leckstroms an der Gesamtleistung gering. Die Nutzung von Clock Gating führt dazu, dass im Ruhezustand kaum dynamischer Leis-

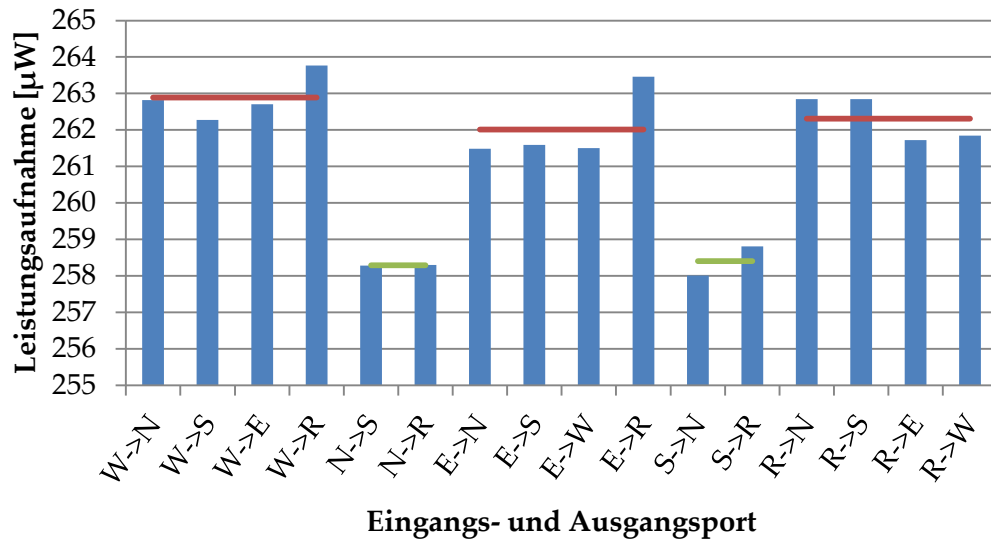


Abbildung 4.8: Dynamische Leistungsaufnahme eines Flits von Eingangs- zu Ausgangsport des Routers – sortiert nach Eingangsport, durchschnittliche Leistungsaufnahme der jeweiligen Ports (Linie)

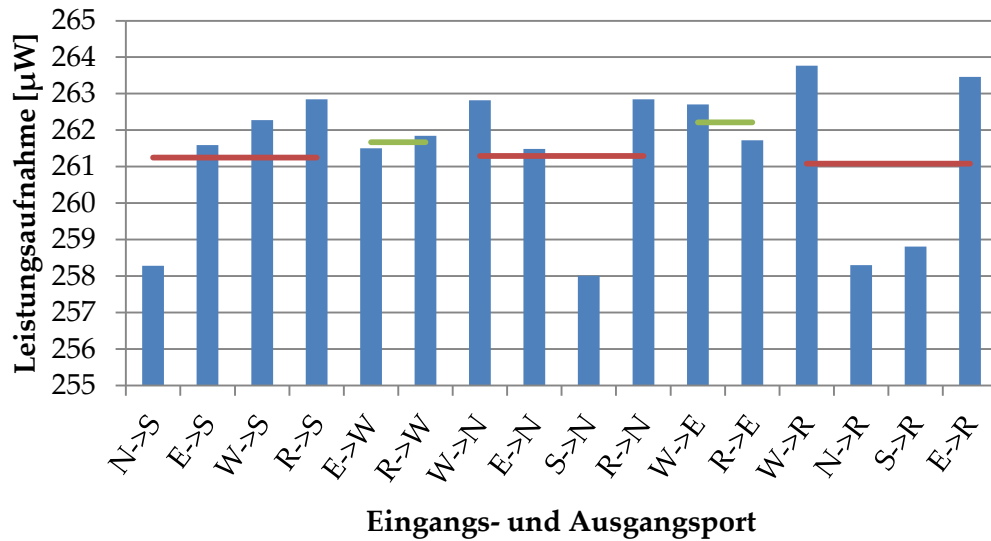


Abbildung 4.9: Dynamische Leistungsaufnahme eines Flits von Eingangs- zu Ausgangsport des Routers – sortiert nach Ausgangsport, durchschnittliche Leistungsaufnahme der jeweiligen Ports (Linie)

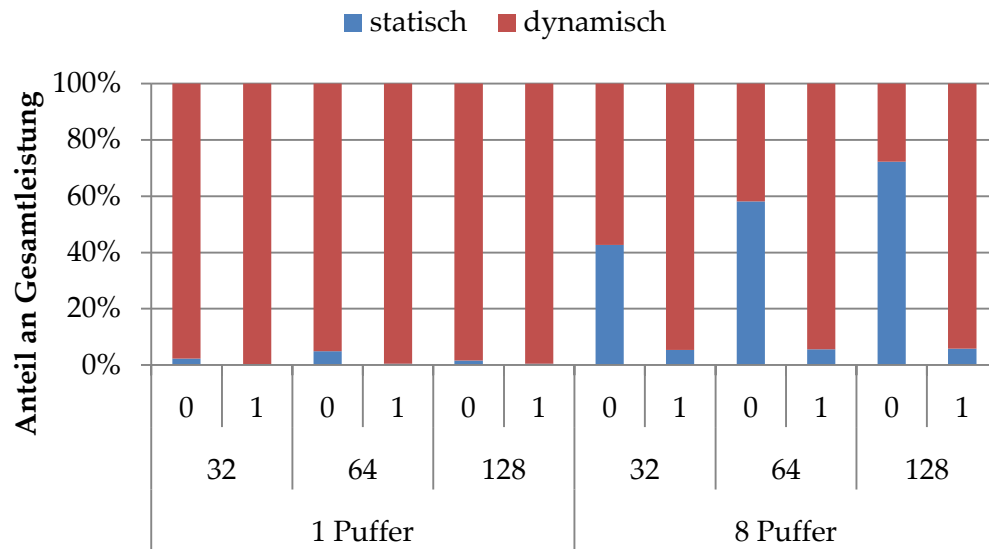


Abbildung 4.10: Anteil des statischen bzw. dynamischen Leistungsumsatzes bei Bearbeitung von 0 (Ruhezustand) und 1 Flit (65 nm, CG-R), für 32, 64 und 128 Bit sowie die Puffertiefen 1 und 8

tungsumsatz stattfindet. Deshalb kann in diesem Fall der Leckstrom bei besonders großen Router-Varianten und niedrigen Frequenzen sogar überwiegen (siehe Abbildung 4.10).

Wie in Abbildung 4.11 zu erkennen ist, steigt die statische Leistungsaufnahme mit der verwendeten Bitbreite und der Puffertiefe ungefähr linear an. Bei den Varianten mit 128 Bit und 4 sowie 8 Pufferstufen liegt der Leckstrom etwas höher als erwartet. Dies wird durch die höheren Treiberstärken, die auf dem kritischen Pfad verwendet werden müssen, um die geforderte Performance zu erreichen, verursacht. Noch stärker zeigt sich dieser Einfluss bei höheren Betriebsfrequenzen. Hier tritt dieser Effekt auch schon bei geringeren Bitbreiten auf.

#### 4.6.1 Interpolation der Parameter

Bei der bisher vorgestellten Methodik die Leistungsaufnahme zu ermitteln, führen Veränderungen der Parameter Bitbreite, Puffertiefe und Zieltechnologie jeweils zu einer erneuten Synthese des NoC-Routers. Daraufhin, wie auch im Fall von Änderungen an der Frequenz, ist eine Simulation zur Bestimmung der Schaltaktivitäten nötig. Dieser



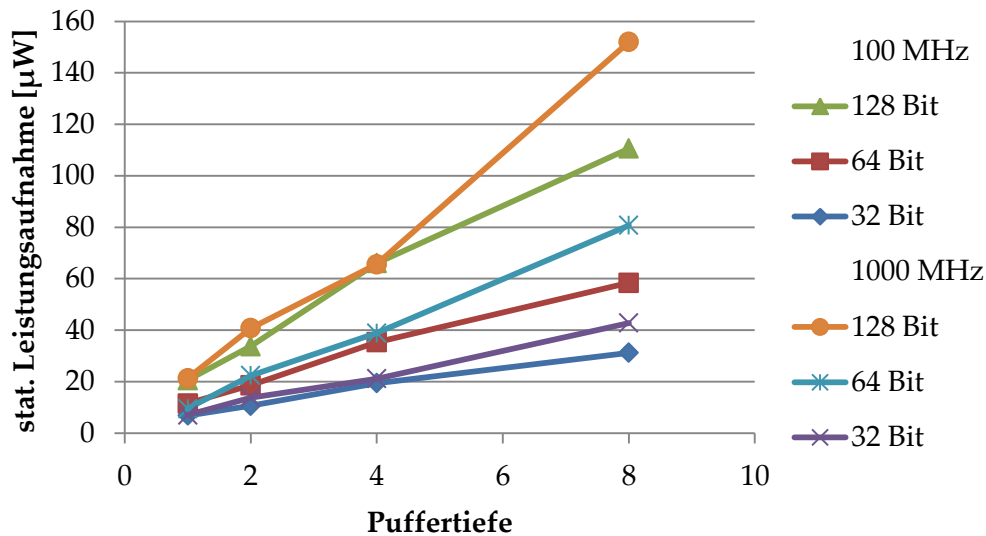


Abbildung 4.11: Statische Leistungsaufnahme der für 100 MHz und 1000 MHz synthetisierten Router

zusätzliche Aufwand macht eine Erhöhung der Analyseauflösung unattraktiv. Eine Alternative bietet das Interpolieren der Parameter. So können unter Umständen bei nur zwei bekannten Ergebnissen weitere Zwischenwerte erzeugt werden. Um die Plausibilität dieser Methode zu untersuchen folgen in diesem Abschnitt einige Auswertungen der Abhängigkeiten von den einzelnen Parametern.

Der Zusammenhang zwischen Betriebsfrequenz und dynamischer Leistungsaufnahme besteht nach Gleichung 2.2 in linearer Weise. Dies kann durch die während der Synthese ermittelten Ergebnisse zum NoC-Router bestätigt werden (siehe Abbildung 4.12). Zum Erreichen höherer Betriebsfrequenzen müssen während der Synthese unter Umständen Zellen größerer Treiberstärke eingesetzt werden. Deshalb kann es dazu kommen, dass nahe der maximal erreichbaren Frequenz – bestimmt durch Fertigungstechnologie und Architektur – eine überproportionale Steigerung des Leistungsbedarfs auftritt. In diesem Fall werden nicht nur die Frequenz sondern auch die Lastkapazitäten geändert. Der gezeigte lineare Zusammenhang gilt deshalb jeweils nur für eine Synthesekonfiguration. In den Untersuchungen zeigte sich jedoch, dass es nur in der Nähe der Grenzfrequenz zum überproportionalen Anstieg des Leistungsbedarfs kommt (siehe Abbildung 4.11).

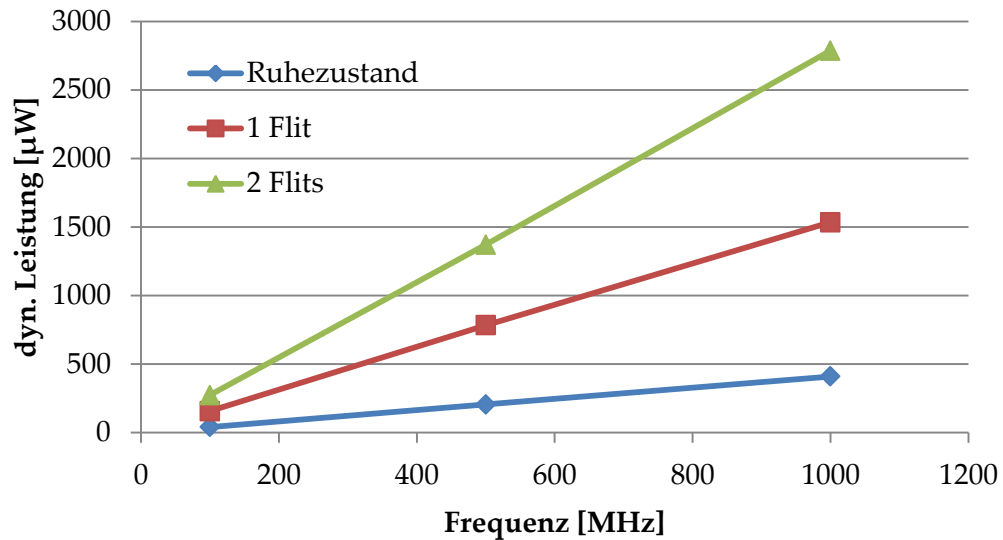


Abbildung 4.12: Abhängigkeit der dynamischen Leistungsaufnahme von der gewählten Betriebsfrequenz

Eine Erhöhung der Bitbreite der die Router verbindenden Links hat eine Vergrößerung der Anzahl der Register zur Folge. Weiterhin wird die Verschaltung über die Crossbar komplexer, da mehr Multiplexer und Demultiplexer benötigt werden. Die Routing- und Arbiterlogik des Routers ist nicht betroffen und bleibt unverändert. Da die Leistungsaufnahme des Routers von der Größe des Pufferspeichers dominiert wird, ergibt sich wiederum eine lineare Abhängigkeit der Leistungsaufnahme. Dies kann durch die vorliegende Untersuchung (siehe Abbildung 4.13) bestätigt werden. Es ist ebenfalls zu erkennen, dass der Leistungsumsatz im Ruhezustand mit steigender Link-Breite nicht zunimmt. Dieser bleibt nahezu konstant, da die statische Leistungsaufnahme nur einen sehr kleinen Teil der Gesamtleistung ausmacht.

Die Erhöhung der Puffertiefe sorgt für eine Vervielfachung der Register. Die Router- und Arbiterlogik sowie die Crossbar bleiben unverändert. Es ist wiederum eine lineare Abhängigkeit anzunehmen. Wie in Abbildung 4.14 zu erkennen, ist im Ruhezustand keine Änderung der dynamischen Leistungsaufnahme feststellbar. Im Fall von Schaltaktivität auf den Router-Ports besteht ab einer Puffertiefe von zwei der erwartete lineare Zusammenhang. Der Fall eines einstufigen FIFO-Speichers stellt insofern eine Besonderheit dar, als dass hier keine Logik zur Bestimmung der letzten Lese- und Schreibposition im

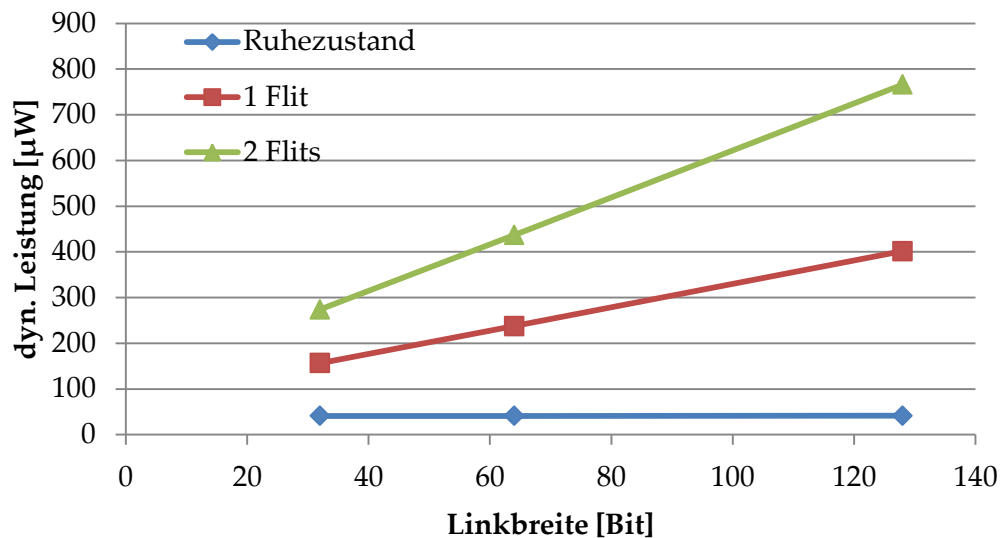


Abbildung 4.13: Abhängigkeit der dynamischen Leistungsaufnahme von der gewählten Bitbreite der Links

Speicher benötigt wird. Diese Einsparung sorgt für eine geringere Leistungsaufnahme als bei Verwendung von mehr als einer Speicherstufe.

Die verwendete Technologie kann nicht ohne weiteres interpoliert werden, da viele der Fertigungsparameter nicht einfach skaliert werden können und sich demzufolge auch keine skalierbare Leistungsaufnahme ergibt. Für die anderen veränderbaren Parameter können bis auf einige Ausnahmen, wie die unmittelbare Nähe zur höchsten erreichbaren Frequenz, lineare Interpolationen durchgeführt werden.

Der Router kann bis zu fünf Pakete gleichzeitig weiterleiten. In diesem Fall sind alle Ports ausgelastet. Um alle möglichen Lastsituationen zu simulieren, müssten sehr viele Kombinationen untersucht werden. Die Leistungsaufnahme im Ruhezustand stellt einen Grundpegel dar. Mit jeder zusätzlich aktivierten Logik wird ein gewisser Teil an zusätzlichem Leistungsumsatz generiert. Wie in den vorangegangenen Abbildungen ersichtlich wurde, steigt der Leistungsbedarf des Routers linear mit der Auslastung weiterer Ports. Aus diesem Grund ist es im Sinne einer Vereinfachung zulässig den Leistungsumsatz während des Ruhezustands und während der Nutzung eines Ports zu ermitteln. Aus diesen zwei Werten kann für jede Anzahl aktiver Ports die Leistungsaufnahme nach

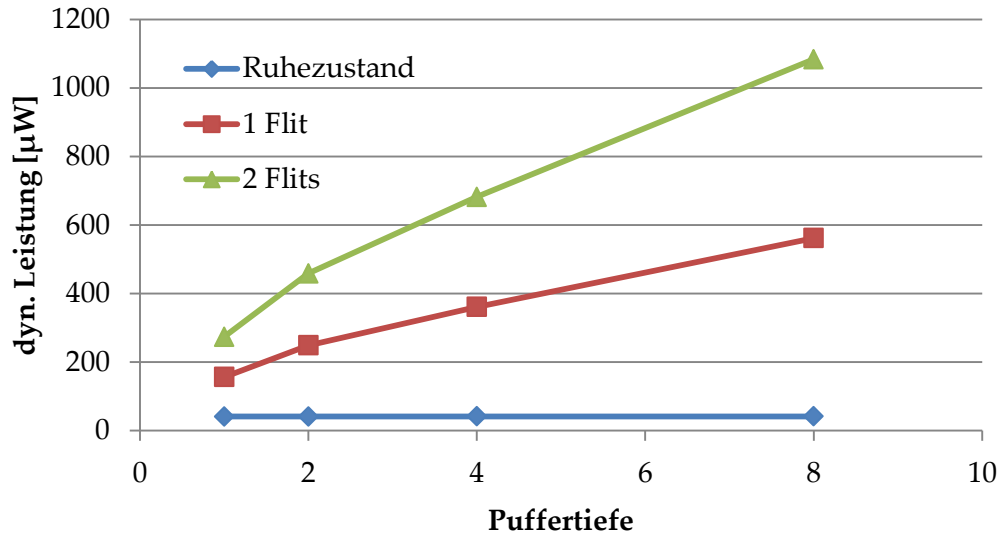


Abbildung 4.14: Abhängigkeit der dynamischen Leistungsaufnahme von der gewählten Puffertiefe des FIFO-Speichers

Gleichung 4.1 geschätzt werden. Der durch diese Extrapolation induzierte Fehler liegt im Bereich von bis zu 2 % wie im Falle der verschiedenen Ports (siehe Kapitel 4.6).

$$P = P_{Ruhe} + n \cdot (P_{1Flit} - P_{Ruhe}) \quad (4.1)$$

#### 4.6.2 Vergleich der Ergebnisse

Die Ergebnisse, die durch die Synthese gewonnen wurden, können nun mit bestehenden analytischen Methoden verglichen werden. Zu diesem Zweck werden das Design Space Exploration for Network Tool (DSENT) [SCK<sup>+</sup>12] und das Multicore Power, Area, and Timing Framework (McPAT) [LAS<sup>+</sup>13, LAS<sup>+</sup>09] genutzt. Während DSENT explizit auf die Abschätzung der Designparameter von NoCs abzielt, können mit McPAT ganze Systeme aus ein oder mehreren NoCs und Prozessoren evaluiert werden.

DSENT ist eine Weiterentwicklung von ORION [WZPM02, KLPS09]. Es besteht aus einem hierarchischen Modell, das zum Teil auf früheren Methoden von Wattch [BTM00] und CACTI [WJ96] basiert. Hinzu kommt eine Korrektur der Skalierungsgrößen unter Berücksichtigung von aktuellen Standardzellbibliotheken industrieller Hersteller. Aus

Grundbausteinen, wie Multiplexer, Arbitr, Crossbar und Pufferspeicher wird je nach gewählten Parametern ein Router- und Link-Modell erstellt. Aus der Summe der Einzelbeiträge ergeben sich dann Daten für ein Mesh-Netzwerk. DSENT unterstützt außerdem die Abschätzung optischer On-Chip-Übertragungsverfahren. Das Ziel dieses Tools ist, den Flächenbedarf sowie die Energie- und Leistungsaufnahme des NoC abzuschätzen. Eine Datenabhängigkeit des Energieumsatzes wird nur in dem Sinne widerspiegelt, als dass sich eine Auslastung bzw. Injektionsrate für das gesamte NoC angeben lässt.

Einen Schritt weiter geht das Tool McPAT. Es basiert ebenfalls auf CACTI, ist jedoch darauf ausgelegt neuere Fertigungstechnologien zu unterstützen und moderne Architekturen wie Multicore und Clustering zu explorieren. McPAT modelliert ganze Prozessorsysteme – inklusive mehrerer Prozessorkerne, Caches, Speichercontroller und hierarchischer NoCs. Diese Grundelemente lassen sich in bestimmten Eigenschaften und Implementierungsoptionen parametrisch anpassen. McPAT erzeugt mit Hilfe seiner Modelle Ergebnisse für jedes der Subsysteme, die den Flächenbedarf, die dynamische, statische sowie Kurzschlussleistung und die erreichbaren Taktfrequenzen umfassen. Die Modelle basieren dabei auf den ITRS-Roadmaps und sind somit auch auf neuere Technologien und Änderungen der Skalierungsvorhaben eingestellt.

Zum Vergleich mit den in den obigen Abschnitten gewonnen Erkenntnissen müssen die Tools mit Hilfe der angebotenen Parameter möglichst auf die hier verwendete Architektur eingestellt werden. DSENT bietet mehrere Optionen für die Crossbar und die Pufferspeicher an. Damit lässt sich die verwendete Architektur mit Multiplexern und Registern nachbilden. McPAT verwendet eine Matrix-Crossbar mit Tri-State-Buffer. Als Pufferspeicher wird SRAM modelliert. Außerdem wird eine mehrstufige Pipeline verwendet, die den Router sehr schnell macht, aber mehrere Taktzyklen Latenz einfügt. Diese Unterschiede lassen sich nicht durch die Konfiguration von McPAT ändern, sondern würden Änderungen an den genutzten Modellen voraussetzen. Deshalb werden auch Differenzen in der Leistungsaufnahme erwartet.

Beim Vergleich eines Routers mit einer Puffertiefe von 1, 2 und 4 sowie Bitbreiten von 32, 64 und 128 Bit in 32 nm-Technologie (siehe Abbildung 4.15) zeigt sich dann auch, dass die Abschätzung bei McPAT 2 bis 36 mal höher als die vorliegenden Synthese-Ergebnisse sind. Vor allem bei hohen Bitbreiten zeigt sich, dass die Matrix-Crossbar in Verbindung mit den Pipeline-Stufen eine höhere Leistungsaufnahme verursacht. Die Puffertiefe der SRAM-FIFO hat hingegen nur einen sehr kleinen Einfluss auf die Leistungsaufnahme des

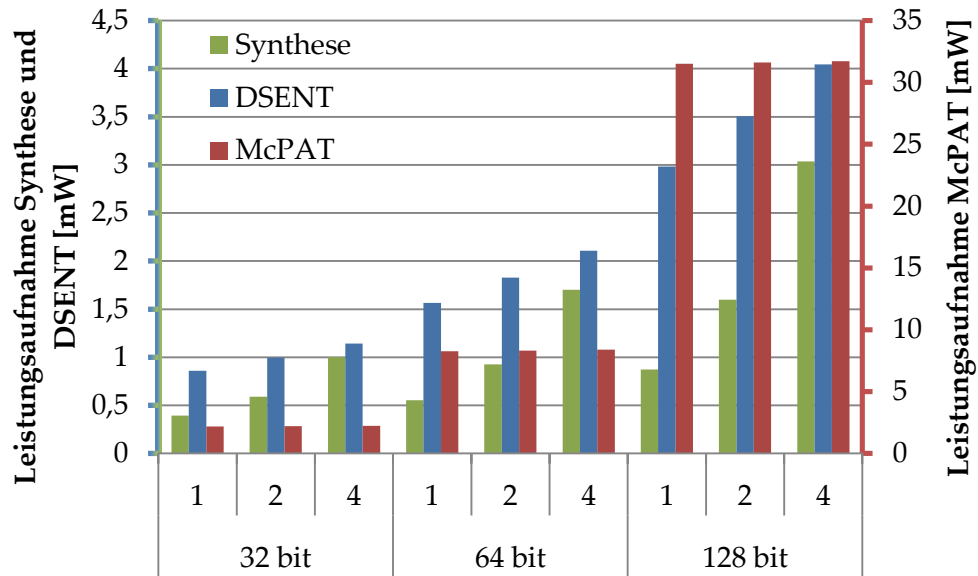


Abbildung 4.15: Vergleich der Leistungsaufnahme für einen NoC-Router aus den Synthese-Ergebnissen und analytischen Abschätzungen von DSENT und McPAT

gesamten Routers. Die Ergebnisse von DSENT zeigen vergleichbare Ergebnisse zur Synthese, wobei die größten Abweichungen bei hohen Bitbreiten entstehen und mit steigender Puffertiefe abnehmen. In allen untersuchten Verfahren lässt sich die annähernd lineare Steigerung der Leistungsaufnahme mit erhöhter Puffertiefe sowie Bitbreite nachweisen.

Im Vergleich von DSENT, McPAT und dieser Arbeit (siehe Tabelle 4.4) wird klar, dass hier bisher nur die Kommunikationsinfrastruktur des Systems modelliert wurde. Das Zusammenspiel des NoC mit Prozessorkernen bildet jedoch einen zentralen Einfluss der Bewertung der unterschiedlichen NoC-Parameter. Aus diesem Grund wird im folgenden Kapitel das NoC-Modell in ein Systemmodell eingefügt. Die einzelnen Cores werden in McPAT ebenfalls modelliert. Dies wird an dieser Stelle nur auf einer groben Detailstufe erfolgen, da der Fokus auf der Bewertung und Modellierung des NoC liegen soll.

Während bei DSENT verschiedene Implementierungen der NoC-Router – vor allem in der Frage der Architektur von Crossbar und Puffer-Speicher – wählbar sind, ist McPAT auf eine festgelegte Router-Architektur beschränkt. Beide sind jedoch durch Skalierung und das Beachten der ITRS-Roadmap in der wählbaren Technologie flexibel. Der in dieser

Tabelle 4.4: Vergleich von DSENT, McPAT und dem Ansatz dieser Arbeit

	DSENT	McPAT	diese Arbeit
Modellierungsziel	NoC	gesamtes System	?
Grundlegende Modelle	ITRS, Zellbibliotheken	ITRS, CACTI	Synthese, Standardzellen
Crossbar und Puffer Architektur	wählbar	fest	fest
Flexibilität	hoch	mittel	niedrig
erwartete Genauigkeit	niedrig	niedrig	hoch
Datenabhängigkeit	Auslastung	Auslastung	?

Arbeit gewählte Ansatz der synthesebasierten Evaluation der NoC-Komponenten sorgt für weniger Flexibilität in diesen Punkten, verspricht dafür jedoch eine höhere Genauigkeit.

Die Leistungsaufnahme aller Komponenten ist von ihrer Auslastung bzw. den anfallenden Kommunikationsvorgängen und sogar der Schaltaktivität innerhalb der Pakete abhängig. Im Falle von DSENT und McPAT können ohne Erweiterung der eigentlichen Tools nur Auslastungen der einzelnen Komponenten angegeben werden. Dies ermöglicht Aussagen zur Leistungsaufnahme bei durchschnittlichem Verkehrsaufkommen innerhalb des NoCs. Auch die Schaltaktivität innerhalb der einzelnen Pakete kann nur im Mittel berücksichtigt werden. Zur Beurteilung des Systems bzw. NoCs wäre es wünschenswert, eine genauere und an die jeweilige Software-Anwendung angepasste Auslastung der Komponenten zu integrieren. Eine solche Option wird ebenfalls im nächsten Kapitel genauer beschrieben.

## 4.7 Fazit

In diesem Kapitel wurde zunächst ein NoC-Router entworfen, der in einem 2D-Mesh mit XY-Routing eingesetzt werden kann. Nach einigen Verbesserungen – wie der Einfüh-

rung von Clock-Gating – konnte die Effizienz der Architektur in Fläche, Performance und Leistungsumsatz nachgewiesen werden. Die Ergebnisse für die Leistungsaufnahme wurden mit analytischen Abschätzungen der zwei bekanntesten Tools für diesen Zweck verglichen. Da die Synthese der verschiedenen Router-Konfigurationen einen großen Aufwand kostet, wurde untersucht, inwieweit sich die Ergebnisse aus einigen wenigen Synthese-Durchläufen interpolieren lassen. Für eine konkrete Technologie müssen auf diese Weise nur sehr wenig Ergebnisse vorliegen, um auf andere Konfigurationen mit verschiedenen Puffertiefen und Bitbreiten zu schließen.

In Anbetracht dieser Erkenntnisse stellt sich eine gewisse Methodik als geeignete Möglichkeit heraus, die Leistungsaufnahme des NoC in einer Systemsimulation abzuschätzen. Zunächst müssen einige Synthesevorgänge des NoC-Routers für jede mögliche Zieltechnologie durchgeführt werden. Eine folgende funktionale Simulation auf Gatterebene ermittelt die nötigen Stützpunkte für die oben genannten Interpolationsmöglichkeiten der Parameter. Dieses Vorgehen ermöglicht eine Charakterisierung der zu nutzenden Hardware und die Rückannotation der Leistungsaufnahme für Simulationen auf höheren Entwurfsebenen.

Konkret (siehe Abbildung 4.16) wird die Methode in dieser Arbeit mit einem NoC-Router durchgeführt, der unter Zuhilfenahme der genannten drei Technologiebibliotheken synthetisiert wird. Mit Hilfe eines kurzen Testszenarios wird ermittelt, welche Schaltaktivität für kein (Ruhezustand) und ein gleichzeitig bearbeitetes Flit festgestellt werden kann. Die korrespondierende Leistungsaufnahme wird mit Hilfe der annotierten Zellbibliotheken ermittelt. Die entsprechenden Werte werden nun in einer Tabelle vorgehalten und können für die oben genannten Interpolationen genutzt werden. Dieses Leistungsmodell wird genutzt, um während der Systemsimulation mehrere Router, also ein ganzes NoC zu annotieren. So können in kurzer Zeit verschiedene Anwendungen und Architekturen getestet werden. Letztendlich lassen sich Aussagen zur Performance der Kommunikationsarchitektur, der Leistungsaufnahme sowie der resultierenden Temperaturverteilung gewinnen.



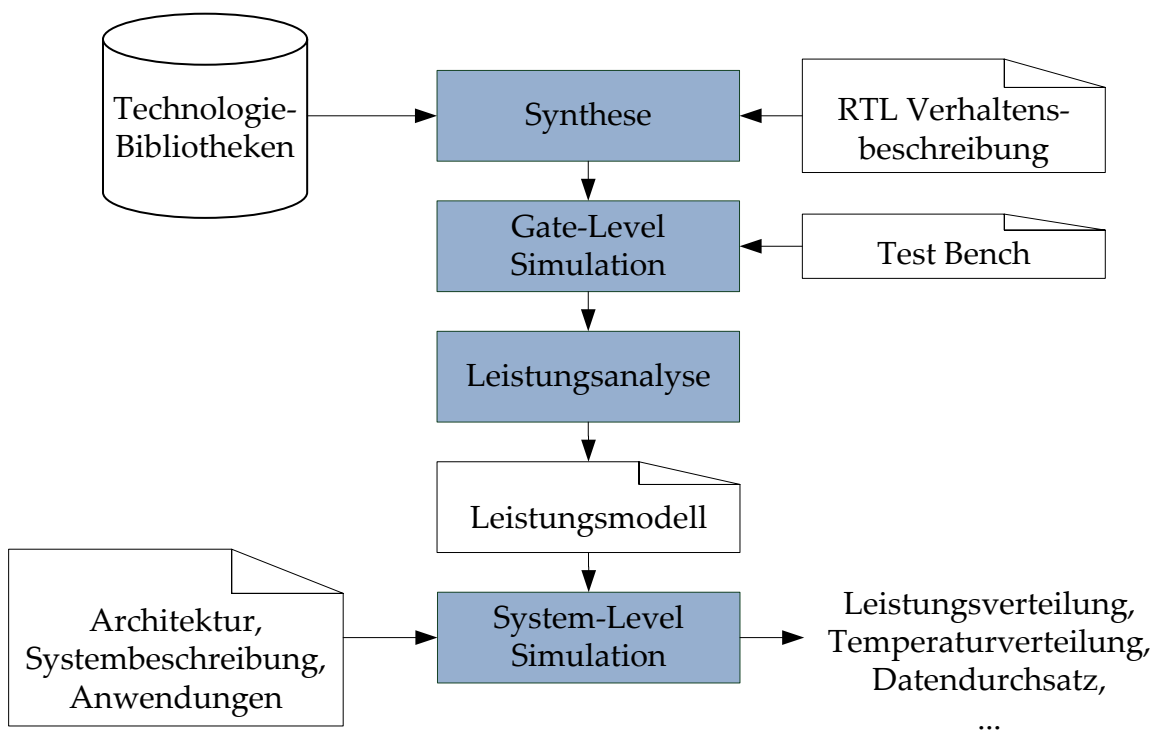


Abbildung 4.16: Grundlegende Methodik, die Ergebnisse zum Leistungsumsatz aus der Simulation auf Gatterebene auf Systemebene zu nutzen



## Kapitel 5

### NoC-Modell auf Systemebene

Um die erworbenen Erkenntnisse über die Router-Implementierungen auf höheren Ebenen anwenden zu können, wird ein NoC-Modell auf Systemebene genutzt und um einige Funktionen erweitert. Einerseits soll eine schnelle Simulationsumgebung entstehen, die den Datenverkehr eines Many-Core-Prozessors unter verschiedenen Architekturparametern abbilden kann. Andererseits sollen Energie- und Leistungsaufnahme sowie die entstehende Temperaturverteilung innerhalb des Systems abgeschätzt werden.

Der gewählte NoC-Simulator wurde am Institut für Angewandte Mikroelektronik und Datentechnik in SystemC umgesetzt [Pfe09, GWG<sup>+</sup>13] und realisiert ein Netz aus 5-Port-Routern. Grundsätzlich ist eine normale Mesh-Topologie, wie auch eine an den Rändern erweiterte Mesh-Variante (Border Enhanced Mesh – BEAM [Kub11]) wählbar. Die BEAM-Topologie zeichnet sich dadurch aus, dass die Router am Rand des Mesh anstatt einer bis zu drei Ressourcen anbinden können. Dazu werden die Ports, an denen normalerweise weitere Router folgen würden für den Anschluss von weiteren IP-Cores genutzt. Da gerade im Falle von XY-Routing ein gehäuftes Verkehrsaufkommen in der Mitte des Netzes zu erwarten ist, bringt die BEAM-Topologie neben der höheren Router-Auslastung eine bessere Verteilung des Verkehrs mit sich.

Sollte die normale Mesh-Topologie gewählt werden, ist an jeden Router eine Ressource angeschlossen. Diese wird auch als IP-Core bezeichnet. Die Funktion dieser IP-Cores wird zunächst nicht modelliert. Stattdessen werden verschiedene Möglichkeiten zur Verkehrssimulation zur Verfügung gestellt. Die einfachste Variante besteht darin, jeden IP-Core in einem zufälligen zeitlichen Abstand ein zufällig langes Paket generieren zu lassen.

Die Zufallsparameter sind einstellbar. Diese Form von synthetischem Verkehr ist dazu geeignet allgemeine Erkenntnisse zur Performance des NoC zu erreichen. Realistischen Anwendungsszenarien entspricht dies jedoch nicht.

Weiterhin ist es möglich, Task-Graphen als Grundlage für die Paketerzeugung zu wählen (siehe Kapitel 5.3). Diese Möglichkeit wurde in Teilen im Rahmen von studentischen Arbeiten am Institut für Angewandte Mikroelektronik und Datentechnik integriert [Pas11, GWG<sup>+</sup>13]. Außerdem ist es denkbar, eigene Funktionen z. B. direkt in der Hochsprache C oder über Instruktionssimulatoren in den NoC-Simulator zu integrieren.

Um die ausgewählten Tasks auf einem Many-Core-System zu verteilen, wird ein Mapping-Verfahren benötigt. In diesem Fall wurde eine Methode implementiert, die speziell die Parallelität der Anwendungen berücksichtigt.

Die Integration eines aus den vorangegangenen Betrachtungen hergeleiteten Modells der Leistungsaufnahme und der folgenden Temperaturverteilung im System ermöglicht zum Ende dieses Kapitels eine Anwendung der Simulationsumgebung in Hinblick auf verschiedene Verkehrssituationen. Vor allem wird die Leistungsfähigkeit verschieden großer Systeme bei unterschiedlich stark parallelisierbaren Anwendungen beurteilt. Das zusätzliche Modell der Temperatúrausbreitung wurde in enger Zusammenarbeit am Institut für Angewandte Mikroelektronik und Datentechnik entwickelt [WGT11, WCG<sup>+</sup>10, WGTU11].

Wesentliche Erkenntnisse dieses Kapitels – besonders Inhalte aus den Abschnitten 5.2, 5.3 und 5.7 – wurden bereits in einer eigenen Publikation [GWG<sup>+</sup>13] veröffentlicht.

### 5.1 Bestehende NoC-Modelle

Innerhalb der letzten Jahre wurden einige Simulationslösungen zur Erforschung von NoCs entwickelt und veröffentlicht. Viele davon dienen der Performance-Evaluation der Kommunikationsarchitekturen und der Abschätzung der unterschiedlichen Parameter zur Topologie sowie Dimensionierung des NoC. Beispielsweise seien hier gpNoC-Sim [HAAN<sup>+</sup>07] und HORNET [LRC<sup>+</sup>11] genannt, die jeweils in den Hochsprachen C++ bzw. Java verfasst sind. Einige der bekannten Lösungen erweitern die Evaluation von NoCs um den Aspekt der Leistungsaufnahme, wie z. B. worm\_sim [HM04] und GARNET [AKPJ09], die jeweils frühe Versionen von ORION nutzen.

Simulationsumgebungen, die auf SystemC als Beschreibungssprache setzen, gibt es ebenfalls. Wiederrum wird die Leistungsaufnahme entweder nicht betrachtet (z. B. NIGRAM [JAHG<sup>+</sup>07], OCCM [CGLP04]) oder es wird eine bestimmte Fertigungstechnologie charakterisiert (z. B. NoCSim [BMKC05], NOSTRUM [J<sup>+</sup>00] mit Nos-HPM [PJ06]) bzw. für weitere Untersuchungen auf ORION verwiesen (z. B. NoCTweak [TB12]).

Mit dieser Arbeit vergleichbare Lösungen zur Abschätzung von Leistungsbedarf, Fläche oder erreichbarer Frequenzen benutzen zumeist die in Abschnitt 4.6.2 beschriebenen Tools DSENT und McPAT. Diese bilden eine flexible aber nur eingeschränkt genaue Grundlage ein Mehrprozessorsystem mit NoC zu modellieren.

Das Simulations-Tool Graphite [MKK<sup>+</sup>10] kann Multicore-Systeme mit über 1000 Kernen evaluieren. Die Besonderheit dabei ist, dass sich der Simulator selbst sehr leicht auf mehrere Multiprozessorsysteme aufteilen lässt. So wird eine hohe Simulationsgeschwindigkeit erreicht. Zur Abschätzung der Leistungsaufnahme wird für das NoC auf ORION bzw. DSENT zurückgegriffen. Caches und Prozessor-Cores werden mit Hilfe von McPAT integriert.

Eine Weiterentwicklung auf Basis von Graphite ist der Sniper Multi-Core Simulator [CHE11], der einen größeren Fokus auf die Unterstützung verschiedener CPUs und Instruktionssätze sowie die Ausführung mehrerer Threads und Anwendungen legt. Da Graphite die Basis dieses Tools bildet, ist die Modellierung der Leistungsaufnahme der Prozessorcores und Caches ebenfalls von McPAT bzw. die der Kommunikationsinfrastruktur von DSENT übernommen.

## 5.2 Implementierung des SystemC-Modells

Mit der Hilfe von SystemC können komplexe Kommunikationssysteme beschrieben werden. Vor allem für Bus-basierte Systeme eignen sich die zur Verfügung gestellten Schnittstellen und auf diese Aufgabe zugeschnittenen Coding-Styles. Es wird u.a. die Verwendung der Styles Loosely Timed, Approximately Timed und Untimed vorgeschlagen. Diese verschiedenen Methoden der Implementierung bieten unterschiedliche Abstraktionsstufen und Aussagequalitäten in Bezug auf das Zeitverhalten. In einigen Fällen ist es jedoch sinnvoll von diesen drei vorgeschlagenen Modellierungsstrategien abzuweichen. Um die gewünschte Simulationsgenauigkeit zu erreichen, wurde daher ein spezifischerer Ansatz gewählt. Die entstehende Simulationsumgebung soll zyklengenau sein, was

ebenfalls mit Hilfe von SystemC abgebildet werden kann. Dies bedeutet nicht notwendigerweise, dass ein Absenken der Abstraktion auf RT-Ebene nötig ist, da die internen Funktionalitäten der Systemmodule weitgehend abstrakt, hochsprachig und algorithmisch beschrieben werden können. Die Kommunikationsschnittstelle und das relativ einfache Übertragungsprotokoll können hingegen zyklenakkurat modelliert werden. Dies bedeutet, dass Aussagen zu den Kommunikationsvorgängen innerhalb des NoC mit der zeitlichen Genauigkeit des Taktsignals der NoC-Router möglich sind.

Zur Umsetzung wurden Elemente der SystemC-Erweiterung des Transaction Level Modeling (TLM) genutzt. TLM 2.0 wurde inzwischen in den SystemC-Standard [IEE12] integriert. Unter anderem wurden zur Modellierung Elemente wie Non Blocking Transport, Sockets, Generic Payload und Eventing genutzt. Das Ziel der Umsetzung war es, eine möglichst effiziente zyklenakkurate Simulation zu ermöglichen. Als Transaktion wird im Folgenden die Übertragung eines Flits zwischen zwei Routern betrachtet. Dies ist anstelle eines Ende-zu-Ende- bzw. paketübergreifenden Transaktionsbegriffes, welcher mehrere Zyklen zusammenfassen würde, nötig, um die Zyklengenauigkeit zu erhalten.

Um die Simulation der NoC-Funktionalitäten möglichst effizient zu implementieren, wurde darauf geachtet möglichst alle Daten, die eine Transaktion betreffen, nur per Zeiger zu behandeln, damit keine ganzen Objekte in andere Speicherkontexte kopiert werden müssen. Dieses Grundprinzip wurde mit der Einführung eines Transaction-Pools umgesetzt. Dieser enthält und verwaltet die erstellten Transaktions-Objekte (TAO), die jeweils ein Flit – also einen Teil eines Pakets – symbolisieren. Der Vorteil dieser Vorgehensweise besteht darin, dass kein neuer Speicher alloziert werden muss, wenn ein neues TAO benötigt wird. In diesem Fall wird einfach ein altes Objekt neu initialisiert. Dies spart vor allem bei einer großen Anzahl von Transaktionen Simulationszeit ein.

Ein TAO besteht aus der dem TLM-Standard entlehnten Generic Payload. Diese wird genutzt um Bus-Transaktionen zu modellieren. Sie enthält eine Möglichkeit das Objekt um beliebige Attribute zu erweitern. Diese Generic Payload Extension wird genutzt, um die Zieladresse des Paketes und die Information, ob es sich um das erste oder letzte Flit eines Paketes handelt, zu übertragen. Die so eingebetteten Informationen werden in den Router-Modulen ausgewertet und zur Wegfindung und Arbitrierung in gleicher Weise genutzt wie in der RTL-Implementierung.

Das Verhalten des Transaction-Pools kann konfiguriert werden. So ist es möglich die Simulationsgeschwindigkeit gegen den Speicherverbrauch abzuwägen. Sollte genügend

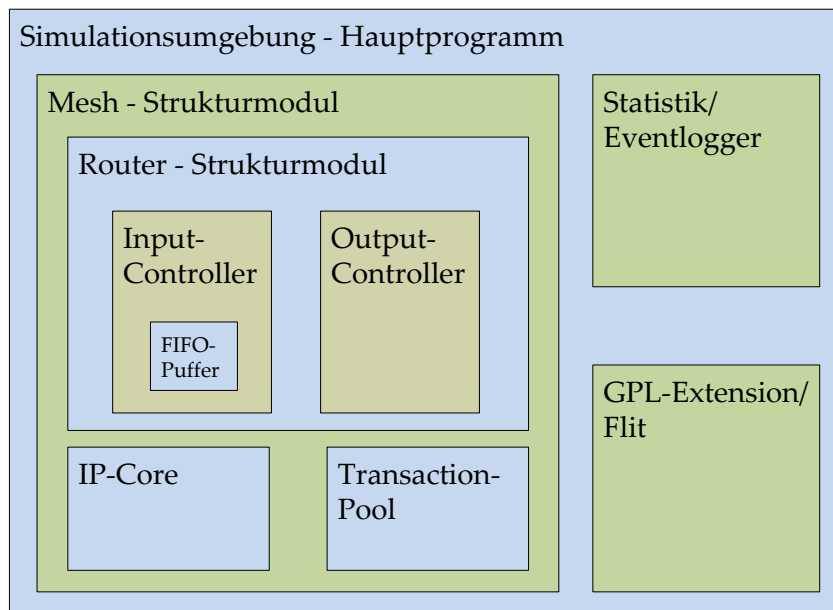


Abbildung 5.1: Grundbestandteile des TLM-Simulators [Pfe09]

Speicher im Simulationssystem zur Verfügung stehen, kann immer die maximal gleichzeitig benötigte Menge an TAO vorgehalten werden. Muss die Speicherbelegung gering gehalten werden, werden Objekte sobald sie nicht mehr benötigt werden freigegeben und bei erneutem Bedarf neu alloziert.

Die Implementierung besteht hauptsächlich aus der Deklaration von einigen Klassen und deren ein- oder mehrfacher Instanziierung (siehe Abbildung 5.1). Das Ausgangsobjekt ist neben einem Statistikmodul und der Definition der Flits das Mesh. Bei Instanziierung des Mesh wird die gewünschte Anzahl von Routern und IP-Cores angelegt. Außerdem erfolgt die Initialisierung des Transaction-Pools, der später die Transaktionsobjekte verwalten wird. Innerhalb eines jeden Routers gibt es je Port einen Input- und einen Output-Controller. Diese bilden die Eingangs- und Ausgangslogik. Der Input-Controller enthält außerdem den FIFO-Puffer.

Einige dieser Objekte entsprechen der Aufteilung, die auch schon für die RTL-Umsetzung genutzt wurde. So gibt es einen Eingangs-Controller und einen Ausgangs-Controller. Im Eingangs-Modul werden die ankommenden Flits behandelt. Zunächst gelangen die TAO über ein Eingangs-Socket in das Eingangs-Modul. Dann erfolgt eine FIFO-Pufferung, wie im Pufferspeicher des RTL-Designs. Dieser FIFO-Puffer besteht aus einem Vektor aus

Zeigern auf die betroffenen TAO. Die Kommunikation mit den darauffolgenden Prozessen erfolgt mittels Eventing. Dadurch wird der entsprechende Thread nicht aufgerufen, so lange keine Daten für ihn vorliegen. Darauf folgt eine Routing-Entscheidung basierend auf der Zieladresse (siehe Tabelle 4.1), die aus dem ersten Flit des Pakets ausgelesen wird. Dann wird das Flit über einen der Ausgangs-Sockets weitergegeben.

An der nächsten Station, dem Ausgangs-Controller, laufen die Eingänge der verschiedenen Ports für jeweils einen Ausgang zusammen. Hier erfolgt eine Round-Robin-Arbitrierung, falls mehrere Pakete gleichzeitig ankommen sollten. In diesem Modul ist eine Zustandsmaschine implementiert, die für jeden Zustandswechsel eine wait-Anweisung über eine Taktperiode enthält. Damit wird die Abhängigkeit vom Systemtakt simuliert und die Taktgenauigkeit sichergestellt.

Die Kommunikation findet zwischen Input- und Output-Controllern, also den Ports der Router und den IP-Cores statt. Die Weiterleitung geschieht über TLM-Sockets. Für das Request- und Acknowledge-Handshaking, welches für das Übertragungsprotokoll benötigt wird, werden TLM-Funktionen genutzt, die einen Hin- und einen Rückkanal der Sockets implementieren.

Gegenüber einer RTL-Implementierung besteht ein enormer Geschwindigkeitsvorteil. Zur Beurteilung der Simulationsgeschwindigkeit wurde eine RTL-Beschreibung in SystemC erstellt, die genau die gleichen Funktionalitäten umsetzt wie die TLM-Implementierung. Der Geschwindigkeitsgewinn hängt u. a. von der Anzahl der Komponenten im NoC, also dessen Größe ab. Außerdem spielt die Injektionsrate, also die Anzahl der Pakete im Netz eine wichtige Rolle. Insbesondere ein geringes Verkehrsaufkommen verstärkt den Vorteil des TLM-Modells, da hier besonders wenig Aktivität auftritt und der Event-Scheduler von SystemC somit selten aufgerufen wird. Im RTL-Modell hingegen werden in jedem Takt alle Module aktiv.

Für geringe NoC-Größen liegt die Simulationszeit des RTL-Modells bis zu 36-mal höher als die des TLM-Modells (siehe Abbildung 5.2). Bei größeren NoCs wurde eine 20- bis 25-mal höhere Geschwindigkeit erzielt. Es wurde eine Injektionsrate von 10 % verwendet. Dies bedeutet, dass in jedem Taktzyklus die Wahrscheinlichkeit eines neu generierten Paketes am IP-Core bei 10 % liegt.

Bei geringeren Aktivitäten wird der Vorteil der TLM-Simulation besonders deutlich. Der Geschwindigkeitsvorteil bewegt sich in diesem Fall um den Faktor 90 (NoC-Größe: 10x10) (siehe Abbildung 5.3). Höhere Injektionsraten als 20 % bringen das NoC in die



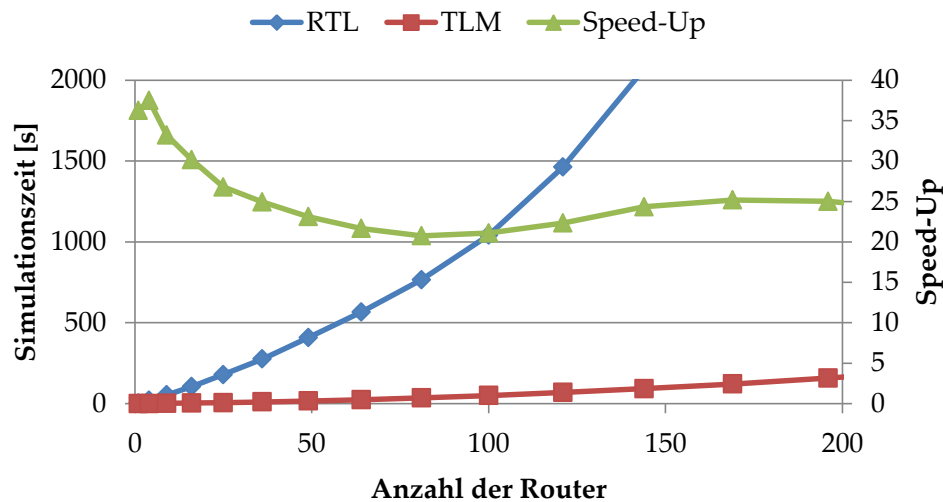


Abbildung 5.2: Simulationsgeschwindigkeit des RTL- sowie des TLM-Modells bei verschiedenen NoC-Größen [TG12]

Sättigung. Das heißt, dass eine Erhöhung der Paketmenge keinen Effekt mehr auf die Auslastung des Netzes hat. Die Simulationszeit des RTL-Modells ist in diesem Fall ca. 15-mal höher.

### 5.3 Erzeugung von Verkehrsmustern

Die zunächst implementierte Variante zur Erzeugung von Datenpaketen im NoC-Simulator ist stark an die eingeschränkten Varianten der RTL-Implementierungen angelehnt. Die Einstellungsmöglichkeiten beschränken sich zunächst darauf eine Injektionsrate und Paketlänge vorzugeben, mit der die Auslastung des gesamten NoC bestimmt werden kann. Da XY-Routing verwendet wird, ergibt sich eine hohe Auslastung zunächst in der Mitte des Mesh, die zu den Rändern hin abnimmt.

Die Umsetzung auf Systemebene bietet weitergehende Möglichkeiten. So müssen umfangreichere Konfigurationsmöglichkeiten geschaffen werden. Jeder IP-Core kann dadurch z. B. mit einer eigenen durchschnittlichen Paketlänge und Injektionsrate versehen werden. Dies impliziert die Möglichkeit unterschiedliche IP-Core-Typen zu modellieren, die unterschiedliche Eigenschaften aufweisen. Weiterhin kann für jeden NoC-Teilnehmer

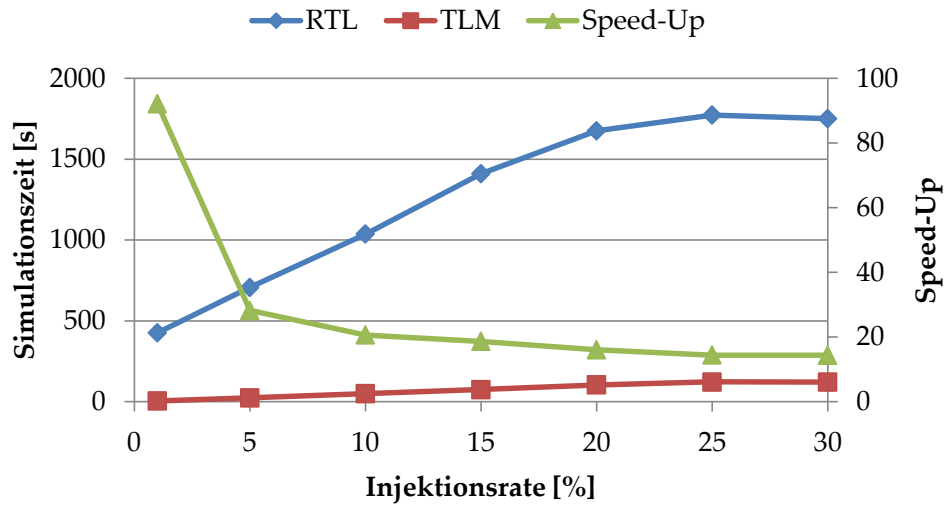


Abbildung 5.3: Simulationsgeschwindigkeit des RTL- sowie TLM-Simulators bei verschiedenen Injektionsraten [TG12]

eine bestimmte feste Zieladresse oder die Zufallsauswahl unter allen gewählt werden. Somit lassen sich einfache Verkehrsmuster für temporäre Situationen im System generieren. Beispielsweise könnte ein IP-Core im NoC einen On-Chip-Speicher oder einen Speichercontroller für Off-Chip-RAM darstellen und Daten generieren, die immer an eine bestimmte Zieladresse geschickt werden. Von dort aus könnte eine serielle oder mehrfach parallele Weiterleitung bis zu einer Datensenke, z. B. dem Ausgabeport oder einem anderen Speichercontroller, modelliert werden. Für eine Änderung dieses Anwendungsbeispiels während der Simulation ist die Konfiguration der einzelnen IP-Cores aber immer noch zu starr.

Um die Verkehrsdatengenerierung weiter zu flexibilisieren wurde die Möglichkeit geschaffen Task-Graphen in den Simulator einzulesen [Pas11]. Die Umsetzung dieses Vorhabens machte eine Änderung bzw. Spezifizierung des Systemmodells nötig. Zunächst wurden IP-Core-Typen eingeführt, die während der Elaborationsphase – also dem Start der Simulation – fest im NoC konfiguriert sein müssen. Um ein heterogenes SoC zu modellieren aber nicht zu spezifisch zu werden, wurden folgende Typen festgelegt:

- Memory/Memory Controller – MC
- General Purpose Processor – GPP

- Digital Signal Processor – DSP
- Application Specific Integrated Circuit – ASIC
- Field Programmable Gate Array – FPGA

Generell handelt es sich bei diesen Typen um datenverarbeitende Module, die sich in ihrer Spezifität unterscheiden. ASIC stellt den spezifischsten Core dar, während die Nutzung eines GPP am variabelsten erfolgen kann. Dabei lassen sich die spezifischen Cores nicht für alle Aufgaben einsetzen, dafür sind sie aber effizienter bei auf sie passenden Tasks. Einen Sonderfall stellt der MC dar. Dieser liefert Instruktions- und Datenpakete an alle Cores aus. Es können mehrere MC im NoC vorhanden sein.

Mit Hilfe einer modifizierten Version von Task Graphs For Free (TGFF) [DRW98] wird ein Set aus mehreren Task-Graphen generiert. Einer der fünf Modul-Typen kann jedem der IP-Cores im NoC zugewiesen werden. Zu jedem Task im Task Graph Set (TGS) ist der zur Ausführung nötige Typ annotiert. Diese Angabe wird bei der Verteilung der Tasks auf die IP-Cores (Mapping) genutzt.

Die generierten Task-Graphen bestehen aus einer Folge von Tasks, die einen oder mehrere Vorgänger und Nachfolger haben. Nur der erste und letzte Task eines Graphen haben keinen Vorgänger respektive Nachfolger. Der Grad der Parallelität und die Dauer der einzelnen Tasks lassen sich konfigurieren und werden im Einzelfall in bestimmten Grenzen zufällig generiert. Des Weiteren bestimmen die Informationen im Task-Graphen wie viele Daten von Task zu Task transportiert und aus MC angefordert werden müssen. All diese Informationen werden im XML-Format von TGFF ausgegeben und in den Simulator eingelesen.

Ein wichtiges Element dieses Vorgangs ist die Zuordnung der Tasks zu bestimmten IP-Cores. Dieser Schritt wird Mapping genannt und bestimmt, wie gut die Parallelität der Tasks innerhalb des Many-Core-Systems ausgenutzt werden kann. Zur besseren Vergleichbarkeit wurden mehrere Mapping-Verfahren implementiert, die verschiedene Eigenschaften des Systems in den Vordergrund stellen.

## 5.4 Task Mapping

Um die Tasks auf dem SoC zu verteilen, ist ein Mapping-Verfahren nötig. Dieses legt nach bestimmten Kriterien fest, welcher IP-Core welchen Task bearbeiten wird. Diese

Entscheidung hat weitreichende Folgen. Zum einen wird die Auslastung und Effizienz des Systems durch die entstehenden Leerlauf-Zeiten der Prozessoren bestimmt. Das Ziel sollte es also sein, eine gleichmäßige Verteilung und eine geringe Gesamtausführungsdauer zu erreichen. Zum anderen spielt bei der Verwendung von Applikationen, die viele miteinander kommunizierende Tasks besitzen, auch der entstehende Kommunikationsaufwand eine große Rolle. Um diesen zu minimieren, muss das Mapping-Verfahren die Anzahl der Kommunikationsvorgänge und die Länge der Wege zwischen den interagierenden IP-Cores gering halten.

Der Ablauf während der Simulation gestaltet sich grundsätzlich so, dass zunächst der erste Task-Graph im TGS auf die zur Verfügung stehenden IP-Cores aufgeteilt wird. Sobald ein IP-Core meldet, dass er alle seine Tasks abgeschlossen hat, wird der nächste Taskgraph platziert. Als Informationen stehen der Mapping-Methode die Beziehungen unter den Tasks, deren Eigenschaften wie Ausführungsdauer und Datenvolumen sowie die bisher zugewiesene Tasks für jeden IP-Core zur Verfügung.

Auf dieser Grundlage wurden mehrere Verfahren implementiert. Die einfachste Variante hat das Ziel, die Anzahl der Tasks pro IP-Core möglichst gleich zu halten. Vor allem während der Evaluierung von Taskgraphen mit parallelen Tasks kann dieses Vorgehen Vorteile haben, da die parallelen Aufgaben auch zeitlich parallel von verschiedenen IP-Cores bearbeitet werden können. Falls jedoch hauptsächlich sequentielle Tasks vorliegen, werden diese auch im kompletten System verteilt und es entsteht ein hoher Kommunikationsaufwand. Unter Umständen kommt es zu größeren Leerlauf-Perioden, da die nachfolgenden Tasks nicht abgearbeitet werden können, solange die Daten der Vorgänger nicht vorliegen.

Eine Strategie, die die Kommunikationspfade minimiert, bietet der „Nearest Neighbour“-Ansatz. Es werden alle Tasks, die voneinander abhängig sind, in direkter Nähe zueinander platziert. Am besten funktioniert dieses Prinzip, wenn alle betroffenen Tasks auf ein und demselben IP-Core laufen. Dies führt allerdings dazu, dass auch parallele Aufgaben auf einem Prozessor laufen sollen. Das ist nicht zeitlich parallel möglich und würde deshalb zu Verzögerungen führen.

Ein weiteres Verfahren stellt das „Load Balancing“ dar. Hier werden die Tasks möglichst gleichmäßig auf alle zur Verfügung stehenden Ressourcen verteilt. Wenn die Abhängigkeiten der einzelnen sequentiellen oder parallelen Strukturen dabei nicht beachtet werden, ergibt sich jedoch eine sehr schlechte Gesamt-Performance des Systems. Z. B. könnten

Tasks mit engen Beziehungen unter Umständen örtlich weit entfernt voneinander platziert werden.

Deshalb wurde eine Methode implementiert, die die Anzahl der parallelen Nachfolger eines Tasks ermittelt und in die Platzierung dieser Nachfolger miteinbezieht. Dafür wird ein quadratisches Gebiet um den Ursprungstask festgelegt, dessen Größe der Anzahl der parallelen Aufgaben entspricht. Die einzelnen Tasks werden in diesem Gebiet verteilt. Damit ist sichergestellt, dass die Kommunikationswege nicht unnötig lang sind und parallele Aufgaben auf verschiedenen IP-Cores abgearbeitet werden. Diese Methode, die eine Mischung aus „Nearest Neighbor“ und „Load Balancing“ darstellt, wird in den nachfolgenden Untersuchungen genutzt.

Darauf aufbauend konnten weitere Mapping-Mechanismen implementiert werden, die auch die Temperaturverteilung innerhalb des Systems berücksichtigen. Dazu wurde das integrierte Temperaturmodell genutzt [WGT11, WGT12, GWG<sup>+</sup>13].

## 5.5 Integration des Verlustleistungsmodells

Die vorangegangenen Betrachtungen zur Leistungsaufnahme des NoC wurden zu einem Modell ausgebaut, welches in die Simulationsumgebung integriert wurde. Zunächst wird der gesamte Betrieb des NoC räumlich in Abschnitte – bestehend aus je einem Router und den ausgehenden Links – aufgeteilt. Die einzelnen Router und ihre Links nehmen je nach Betriebszustand unterschiedlich viel Leistung auf. Um dies zu berücksichtigen, wird der Betrieb in zwei zeitliche Phasen unterteilt.

Eine Möglichkeit ist, dass kein Paket verarbeitet wird. Der Router und die Links befinden sich im Ruhezustand. Die Datenleitungen weisen keine Aktivität auf. Somit tritt nur für die getakteten Schaltungselemente ein dynamischer Leistungsumsatz auf. Bei Einsatz einer Clock-Gating-Technik wird zudem verhindert, dass die Register der Schaltung durch interne Umladungsvorgänge zur dynamischen Leistungsaufnahme beitragen.

Die zweite Phase besteht darin, dass ein Paket geroutet und weitergeleitet wird. Hier tritt zunächst im Router ein Leistungsumsatz auf, der auf die Aktivierung der Pufferspeicher zurückzuführen ist. Wenn die gepufferten Paketeile weitergeleitet werden, werden diese von der Routinglogik verarbeitet. An dieser Stelle spielt es eine Rolle, ob das Flit das erste im Paket (Header-Flit) ist, also die Zieladresse enthält oder es sich um ein Body-Flit handelt, das gemäß Wormhole-Switching einfach über den bestehenden Pfad weitergeleitet wird.

Ein Header-Flit verursacht eine leicht höhere Leistungsaufnahme, da die Wegfindung in der Routing-Logik durchlaufen werden muss. Die zuständige Zustandsmaschine muss dann ein Statusregister beschreiben. Das Body-Flit wird hingegen nur darauf geprüft, ob es das abschließende Flit ist und wird dann über den schon feststehenden Ziel-Port weitergeleitet.

Nun stellt sich die Frage, welche Fälle in welchem Detailgrad durch das Power-Modell unterschieden werden sollen. Einerseits verspricht eine erhöhte Komplexität des Modells ein präziseres Ergebnis, andererseits erhöht jeder zusätzliche Aufwand die benötigte Simulationszeit.

Wie in Kapitel 4.6 gezeigt wurde, haben unterschiedliche Faktoren verschiedene Einflüsse auf die Leistungsaufnahme des Routers. In der Ruhephase ist das vor allem das Vorhandensein eines Clock-Gating-Mechanismus. Der dadurch verursachte Unterschied in der Leistungsaufnahme im Ruhezustand liegt je nach Variante bei bis zu 90 %. Der Unterschied von Header- zu Body-Flit macht bis zu 30 % aus. Außerdem wurde eine Abweichung festgestellt, je nach dem von welchem Eingangs-Port zu welchem Ausgangs-Port die Weiterleitung erfolgt. Dies liegt daran, dass beim verwendeten XY-Routing nicht von jedem auf jeden Port weitergeleitet werden kann. Dadurch sind die Zustandsregister in einigen Ports kleiner als in anderen, was zu einer unterschiedlichen Leistungsaufnahme führt. Die Abweichung beträgt bis zu 2 %.

Aus den genannten Gründen wird während der Systemsimulation zwischen Header- und Body-Flits unterschieden. Die Differenzierung der genutzten Ports wird aufgrund der nur geringen Genauigkeitssteigerung zunächst nicht beachtet. Zur Bestimmung der Leistungsaufnahme wird also die Länge der Ruhephasen und die Anzahl der jeweils gleichzeitig verarbeiteten Flits genutzt. Diese Werte lassen sich anschließend mit den vorher aufgenommenen Werten aus der zellbasierten Leistungssimulation kombinieren.

## 5.6 Integration des Temperaturmodells

Da die Wärmeentwicklung eine direkte Folge des Leistungsumsatzes innerhalb des Chips ist, bietet es sich an diese Auswirkung ebenfalls zu bestimmen, wenn eine derartige Verlustleistungsschätzung vorgenommen wird. Die Wärmeentwicklung ist im Wesentlichen proportional zum Leistungsumsatz am Ort dessen Auftretens. Während die Wärme freigesetzt wird, breitet diese sich innerhalb des Chips aus und fließt in Richtung der niedrigeren

Tabelle 5.1: Dualität zwischen thermischem und elektrischem Modell [TCST10]

Thermisches Modell	Zeichen [Einheit]	Elektrisches Modell	Zeichen [Einheit]
Wärmefluss	$P$ [W]	Strom	$I$ [A]
Temperatur	$T$ [K]	Spannung	$U$ [V]
Widerstand	$R_{th}$ [K/W]	Widerstand	$R$ [V/A]
Kapazität	$C_{th}$ [J/K]	Kapazität	$C$ [As/V]

Temperaturen und somit letztendlich über die Kühlvorrichtung an die Umgebung ab. Diese Verteilung der thermischen Energie kann man unter Zuhilfenahme der Dualität von Thermo- und Elektrodynamik unter Nutzung elektrischer Schaltkreise modellieren. Dabei entspricht ein Wärmefluss dem Auftreten eines elektrischen Stromes und die vorherrschende Temperatur dem Spannungspegel des Knotens. Thermischer Widerstand und Kapazität können durch ihre elektrischen Gegenstücke modelliert werden (siehe Tabelle 5.1).

Aus diesen Erkenntnissen wird nun ein Modell generiert, welches das 2D-Mesh eines NoC auf ein Kapazitäts- und Widerstandsnetzwerk abbildet (siehe Abbildung 5.4). Der Strom, der jeweils an den aktiven Knoten injiziert wird, entspricht proportional dem Leistungsumsatz, der in dem entsprechenden Gebiet des Systems umgesetzt wird. Das sich nach einem gewissen Simulationszeitraum einstellende Spannungsniveau an den einzelnen Knoten kann als Temperaturverteilung innerhalb des Systems interpretiert werden.

Für die prototypische Implementierung in den NoC-Simulator wurde eine Auflösung gewählt, die einzelne Komponenten des NoC als kleinste Einheiten auffasst. Jeder IP-Core, jeder Router sowie jeder Link entspricht einer Zelle aus Widerstand, Stromquelle und Kapazität. Diese ist mit allen Nachbarzellen verbunden. Zusätzlich kann Wärme in die dritte Dimension über Gehäuse, Heat-Spreader und Kühlkörper an die Umgebung abgegeben werden.

Die Simulation des elektrischen Netzwerks erfolgt durch SystemC AMS. Diese Bibliothekserweiterung von SystemC zur Modellierung analoger Systemteile lässt sich für den angestrebten Zweck nutzen und fügt sich gut in die bestehende Simulationslösung ein. Wenn die Schnittstellen zur funktionalen und technologischen Simulation passend ge-

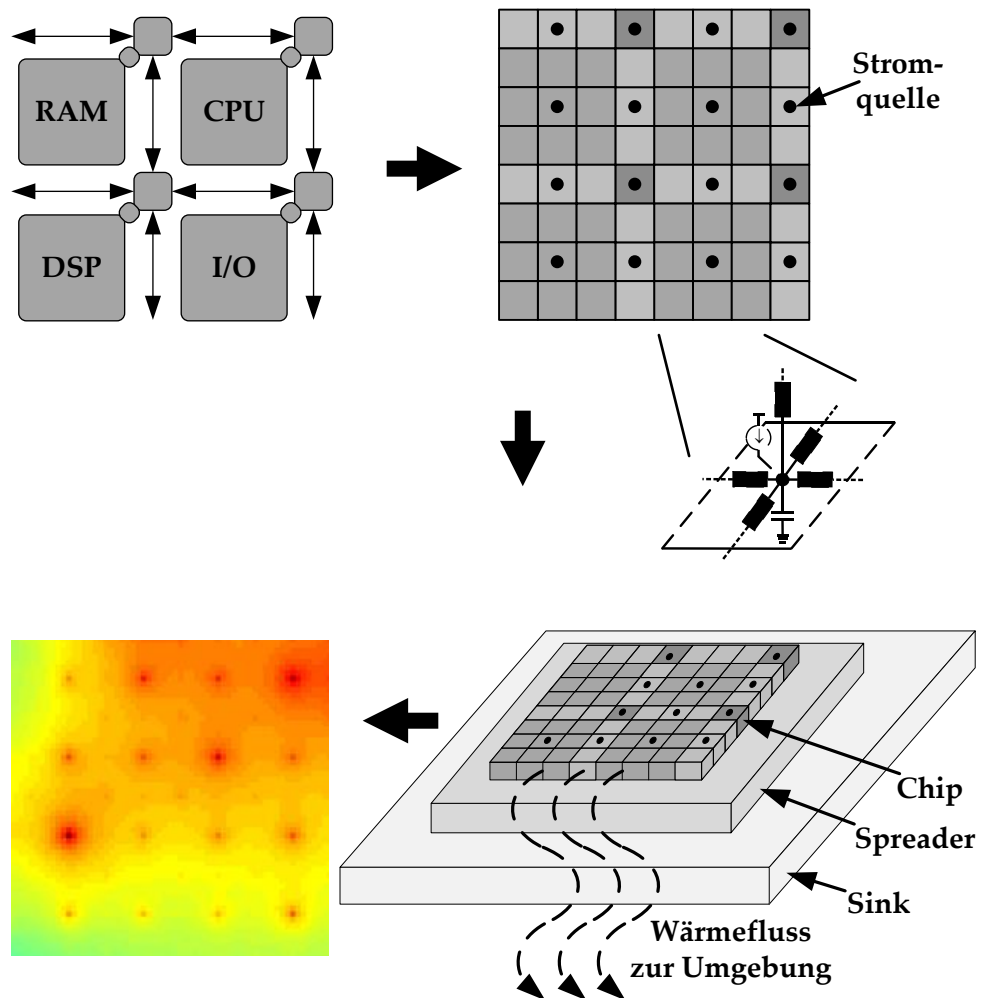


Abbildung 5.4: Umsetzung des Temperaturmodells vom NoC über die Einteilung in aktive und passive RC-Kacheln sowie die Umsetzung in das ELN bis zur resultierenden Temperaturverteilung [WCG<sup>+</sup>10]



wählt werden, wäre es jedoch ohne weiteres möglich, andere Lösungen wie SPICE für die thermale Simulation einzusetzen.

Die Objekthierarchie des Electric Linear Networks (ELN) wird zunächst beim Start der Simulation, sobald alle Parameter bekannt sind, initialisiert. Es folgt die Elaborationsphase, während der das entstandene Differentialgleichungssystem gelöst wird. Während der folgenden Simulation wird in vorher festgelegten Intervallen die Strominjektion je nach berechnetem Leistungsumsatz aktualisiert und die momentanen Temperaturwerte ausgelesen.

Die Strominjektion erfolgt erst nach der Leistungsabschätzung und enthält somit eine zeitliche Verzögerung. Das nächste Auslesen der Temperaturverteilung erfolgt nach dem nächsten Simulationsschritt und kann demnach nicht den neuesten Aktivitäten innerhalb des Systems Rechnung tragen. Die Zeitkonstante für ein Stück Silizium, das eine Partition des Chips darstellen könnte, von 1,0 mm Länge, 0,4 mm Breite und 0,6 mm Dicke [SSH<sup>+</sup>03] entspricht ungefähr 10,95 ms (siehe Gleichung 5.1). Veranschaulicht heißt das, dass ein Temperaturunterschied über ein solches Stück Silizium innerhalb von etwa 6,90 ms ( $\ln 2 \cdot \tau$ ) zu etwa 50 % ausgeglichen ist. Nach dem Verstreichen von fünf Zeitkonstanten betrachtet man den Vorgang gemeinhin als abgeschlossen (Ausgleich zu 99,3 %). Diese Betrachtung kann als grobe Abschätzung gelten. Für genauere Berechnung müssen Differentialgleichungen genutzt werden. Wenn man die Simulationsschritte entsprechend klein wählt, erhält man eine hinreichend aktuelle Temperaturverteilung, um Systemmanagementfunktionen wie Task Mapping oder Skalierung von Betriebsspannung und -frequenz zu evaluieren. Weiterhin ist durch die Kenntnis der thermischen Beanspruchung des Systems eine Abschätzung von Zuverlässigkeitsaspekten möglich.

$$\begin{aligned}
 \tau &= R_{th} \cdot C_{th} \\
 R_{th} &= \frac{1}{150} \frac{\text{m} \cdot \text{K}}{\text{W}} \cdot \frac{1,0 \text{ mm}}{0,4 \text{ mm} \cdot 0,6 \text{ mm}} \\
 C_{th} &= 703 \frac{\text{J}}{\text{kg} \cdot \text{K}} \cdot 2,336 \frac{\text{g}}{\text{cm}^3} \cdot 0,6 \text{ mm} \cdot 1,0 \text{ mm} \cdot 0,4 \text{ mm} \\
 \tau &= 10,95 \text{ ms}
 \end{aligned} \tag{5.1}$$

## 5.7 Anwendung in der Entwurfsraumanalyse

Die Entwurfsraumanalyse eines Systems verlangt die Vergleichbarkeit bei verschiedenen Parameteränderungen. Es ist jedoch selten nur eine einzige Vergleichsgröße für die Auswahl bestimmter Parameter relevant. Eher muss eine geeignete Kombination von benötigter Fläche, Zuverlässigkeit, Leistungsaufnahme und Verarbeitungszeit bzw. Performance gefunden werden. Normalerweise sind für alle diese Größen Grenzen durch äußere Zwänge vorgegeben. Die Kostenkalkulation für ein Produkt kann beispielsweise die maximale Chipfläche beschränken. Das Einsatzszenario bestimmt die minimal benötigte Robustheit und Zuverlässigkeit des Systems und begrenzt eventuell die Bauform des Produktes und damit die maximale Leistungsaufnahme. Der Anwendungszweck verlangt eine bestimmte Performance.

Vor allem wegen des Zusammenhangs zwischen dynamischer Verlustleistung und Frequenz wird innerhalb dieser Grenzen oft Performance gegen Leistungsaufnahme abgewogen (siehe Gleichung 2.2). Diese beiden Größen bieten deshalb einen geeigneten Ansatz generelle Metriken zur Bewertung eines Systems zu entwickeln. Neben der reinen Performance und der Leistungsaufnahme finden auch Kombinationen, wie die benötigte Energie Anwendung. Die benötigte Energie wird auch als Produkt aus Leistung und benötigter Verarbeitungszeit bzw. Power-Delay-Produkt (PDP) beschrieben. Durch die Reduzierung der Betriebsfrequenz lässt sich die Leistung nicht nur linear sondern stärker verringern, solange die Betriebsspannung abgesenkt werden kann. Dies führt dazu, dass nach dem PDP sehr langsame Systeme tendenziell besser sind als schnelle. Um diesen Umstand auszugleichen hat sich die stärkere Wichtung der Performance im Energy-Delay-Produkt (EDP) als weitere Metrik durchgesetzt. Im EDP ist die Zeit quadratisch und die Leistungsaufnahme linear gewichtet. Dadurch schneiden Systeme mit niedriger Frequenz nicht unbedingt besser ab als schnelle Systeme.

Die Performance des hier betrachteten NoC als Kommunikationssystem lässt sich mit dem Anpassen der Parameter Frequenz und Bitbreite direkt steigern. Konkret erhöht sich die Bandbreite pro Link, da mit höherem Takt bzw. mehr Daten pro Takt übertragen werden kann. Weiterhin verringern sich die Latenzen, weil die Übertragung eines Paketes in kürzerer Zeit abgeschlossen wird. Ein Vergrößern der Puffertiefe hat ebenfalls positive Auswirkungen. Dies kommt jedoch nur zum Tragen, wenn die Auslastung des NoC so groß ist, dass Pakete durch andere Pakete blockiert werden. Eine Vergrößerung des Systems durch das Hinzufügen von IP-Cores erweitert das Mesh und steigert die Anzahl

der benötigten Router. In der Folge können parallele Aufgaben besser unter den IP-Cores aufgeteilt werden. Das System wird schneller. Das Kommunikationssystem kann insgesamt gesehen in gleichem Maße eine höhere Bandbreite erreichen. Damit einher geht allerdings auch ein Anstieg der Leistungsaufnahme.

Wegen der Mängel des PDP, wird die Performance häufig höher gewichtet – wie im EDP. Demnach sind größere Systeme durchaus im Vorteil. Der beschränkende Faktor ist die Parallelisierbarkeit der Aufgabe. Die entsprechende – auch als Amdahls Gesetz [Amd67] bekannte – Regel besagt, dass ein System nur bis zu einem bestimmten Punkt von dem Hinzufügen weiterer paralleler Ressourcen profitiert. Ein Teil des zu lösenden Problems ist sequentieller Natur und somit nicht aufteilbar. Das heißt, die Ausführungszeit sinkt ab einem bestimmten Punkt nicht mehr weiter ab.

Zur Überprüfung der bisher aufgestellten Thesen wurden verschieden stark parallelisierbare Anwendungen in der Simulationsumgebung ausgeführt (siehe Abbildung 5.5). Jede der Anwendungen besitzt dabei die gleiche Summe an Einzelausführungsdauern der Tasks. Auch die Anzahl der Teilaufgaben ist gleich. Nur die Aufteilbarkeit in einzelne Threads ist unterschiedlich. In dieser und den folgenden Untersuchungen werden quadratische Systeme mit jeweils zwei bis vier Speichercontrollern verwendet. Die angegebenen NoC-Größen sind also etwas größer als die verfügbaren Recheneinheiten. Es zeigt sich, dass die untersuchten Task-Graphen auf größeren Systemen gleich lang oder schneller abgearbeitet werden als auf kleinen. Die Parallelisierbarkeit kann erst mit mehr zur Verfügung stehenden Ressourcen genutzt werden. Außerdem weicht die theoretisch durch das Mapping und die Systemgröße mögliche Verarbeitungszeit (gestrichelt in Abbildung 5.5) mit zunehmender Systemgröße und Parallelisierbarkeit ab. Da die Tasks mit zunehmender Anzahl von IP-Cores besser verteilt werden können, entsteht ein größerer Kommunikationsaufwand, der sich in einer höheren Wartezeit der Tasks aufeinander niederschlägt. Wenn die Wahl der Systemgröße nur von der Verarbeitungszeit der Tasks abhängen würde, sollte man eine Core-Anzahl wählen, die der möglichen Parallelisierbarkeit, also den gleichzeitig auftretenden Threads entspricht. In diesem Fall reicht ein System mit mehr als zehn IP-Cores für die Anwendung die zehn Threads enthält. Für 20 und 80 gleichzeitige Threads gilt dies analog.

Neben der Performance des Systems für verschiedene Task-Graphen kann auch die Leistungsaufnahme des NoC bestimmt werden. Diese steigt mit zunehmender NoC-Größe. Mehr Aufschluss gibt die Kombination der Verarbeitungsdauer und der Leistungsauf-

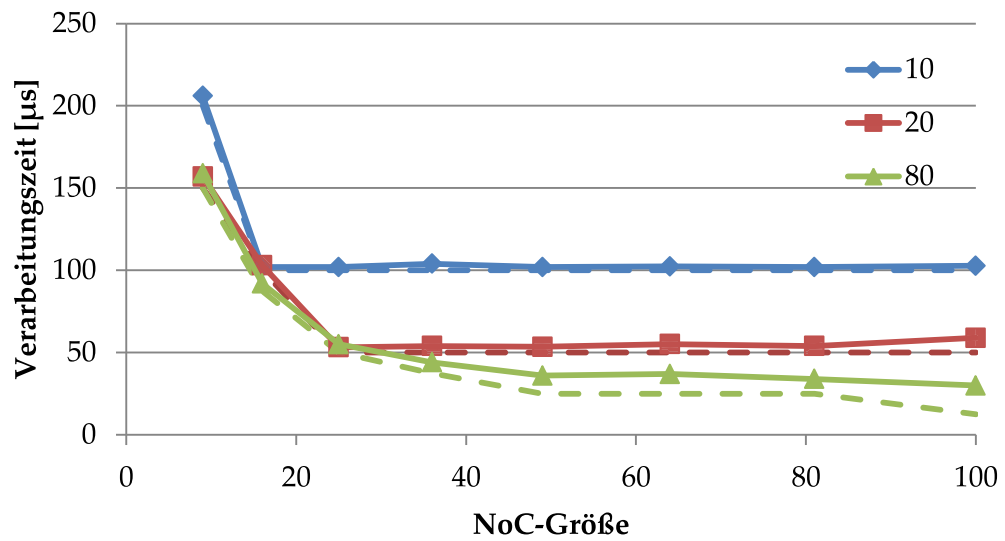


Abbildung 5.5: Ausführungszeit einer Anwendung, aufgeteilt in 10, 20 und 80 parallele Threads auf unterschiedlich großen Systemen quadratischer Topologie – gestrichelte Darstellung entspricht der idealen Verarbeitungszeit ohne Kommunikation

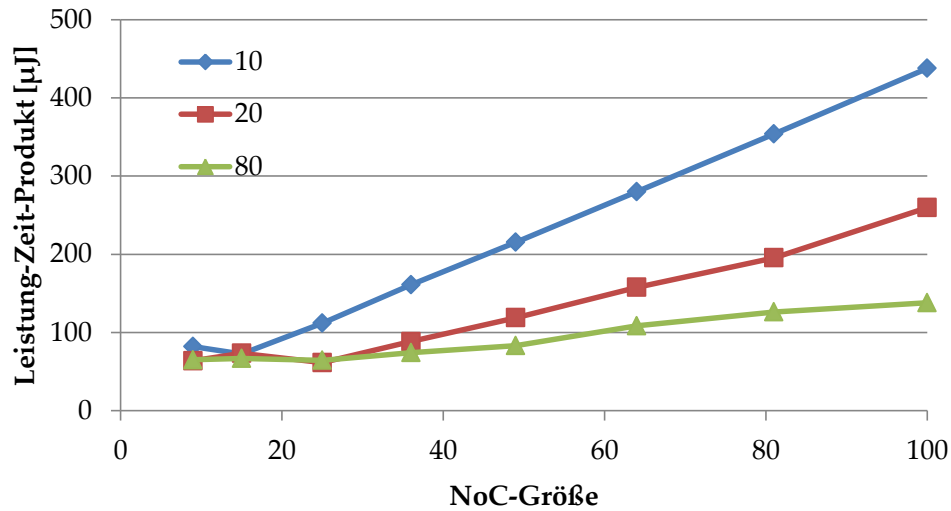


Abbildung 5.6: Energiebedarf verschiedener NoC-Größen bei unterschiedlich parallelen Anwendungen (10, 20, 80 Threads)

nahme zur benötigten Energie. Diese Metrik wird auch als PDP bezeichnet. Da langsame Systeme generell auch weniger Leistungsbedarf besitzen, wird dieses Produkt als Vergleich der Effizienz von Systemen eingesetzt. Hier ist das Bild nicht mehr ganz so eindeutig (siehe Abbildung 5.6). Während die 20-fach parallele Anwendung ihren geringsten Energiebedarf auf dem System mit 16 IP-Cores hat, erreichen sowohl die 20- als auch die 80-fach parallelen Task Graphen ihr bestes Ergebnis bei 25 Kernen. Wäre die benötigte Energie der ausschlaggebende Faktor in der Auswahl der Systemgröße, würde die Wahl also auf 16 oder 25 Kerne fallen.

Da eine CMOS-Schaltung durch Reduzierung des Systemtaktes und Reduzierung der Versorgungsspannung bei linear sinkender Geschwindigkeit eine über das lineare Verhalten hinausgehende Leistungsreduktion erfährt, wird anstelle des Energiebedarfs häufig das EDP als Metrik herangezogen. Während bei den 10- und 20-fach parallelen Anwendungen noch die Verarbeitungsgeschwindigkeit dominiert und die besten Ergebnisse im Größenbereich der besten Parallelisierbarkeit erreicht werden (siehe Abbildung 5.7), steigt das EDP für den 80-fach parallelen Task Graphen bereits ab 49 IP-Cores leicht an. Hier zeigt sich, dass auch bei einer großen Anzahl von IP-Cores und hohem Maß an Parallelisierbarkeit die Effektivität im Sinne des EDP nicht beliebig steigt. Der Kommuni-

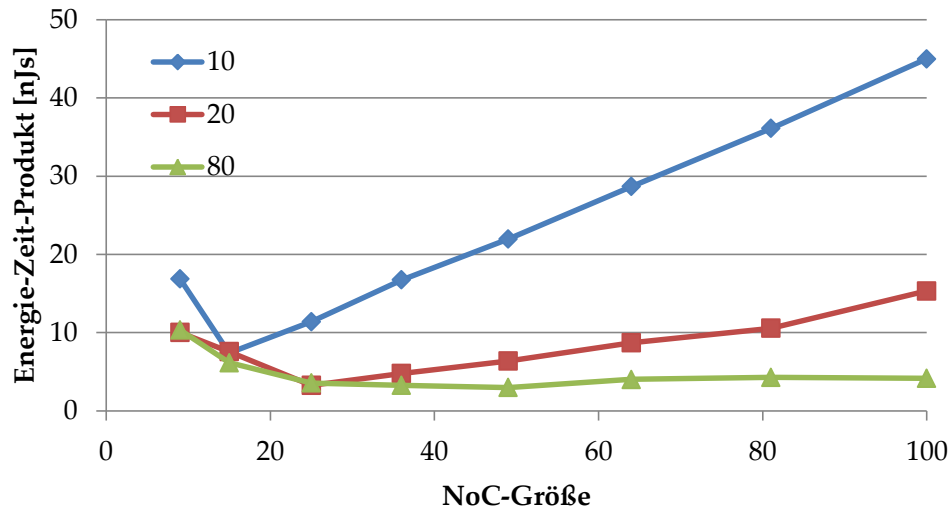


Abbildung 5.7: EDP verschiedener NoC-Größen bei unterschiedlich parallelen Anwendungen (10, 20, 80 Threads)

kationsaufwand übersteigt ab einem gewissen Punkt die Vorteile einer etwas schnelleren Ausführungszeit.

Als beispielhafte Entwurfsraumanalyse sollen zunächst drei verschiedene TGS, die gewisse Applikationen repräsentieren, in Bezug auf ihr EDP untersucht werden. Die untersuchten Anwendungen unterscheiden sich in der Anzahl der enthaltenen Einzel-Tasks, der Dauer eines Tasks und der Menge der Daten, die jeder Task zum Nachfolger transportieren lässt. Außerdem unterscheidet sich der Aufbau der Task-Graphen dahingehend, dass Anzahl und Länge der aus mehreren Tasks gebildeten Threads angegeben werden. Mit diesen Parametern lässt sich die Parallelität innerhalb der einzelnen Task-Graphen vorgeben.

Anwendung 1 (A1) besitzt viele Einzel-Tasks mit kurzer Dauer. Es werden jeweils wenige Daten erzeugt und dem nächsten Task zur Verfügung gestellt (siehe Tabelle 5.2). Anwendung 2 (A2) enthält nur ein Zehntel der Tasks, die dafür aber eine zehnfach längere Dauer haben. Außerdem werden zehn Mal mehr Daten produziert. Der dritte Anwendungsfall (A3) stellt eine Mischung aus den ersten beiden dar. Die Auswahl der einzelnen Parameter innerhalb der vorgegebenen Grenzen pro Task-Graph erfolgt zufällig.

Tabelle 5.2: Verwendete Task-Graph Sets

	# Task-Graphen	# Tasks	Task-Dauer [ns]	Länge/Breite der Threads	Daten [Byte]
A1	50	2000	500	10/200	100
A2	50	200	5000	4/50	1000
A3	50	1000	500-5000	5-500/2-200	100-1000

Um statistische Abweichungen zu verringern besteht jede Anwendung bzw. jedes TGS aus 50 Task-Graphen.

Mit der Länge und der Anzahl von parallel verlaufenden Threads ist die Parallelisierbarkeit der Anwendungen wählbar. A1 besitzt mit 200 nebenläufigen Threads die jeweils 10 Tasks enthalten eine große Parallelität. Mit der großen Menge Einzel-Tasks und den relativ kleinen Datenmengen kann beispielsweise ein System zur Paketverarbeitung eines oder mehrerer Netzwerk-Streams modelliert werden. Im Falle von A2 gibt es nur 50 parallele Threads mit jeweils 4 Tasks. Da hier etwas weniger aber längere Tasks genutzt werden, die auch mehr Daten generieren, kann z. B. ein System für High Performance Computing (HPC) modelliert werden. A3 stellt eine Mischung aus A1 und A2 dar und besitzt dadurch Anwendungen mit mehr oder weniger parallelen Threads. Dieses Anwendungsbeispiel könnte ein Multimedia-SoC, in dem mehrere heterogene Anwendungen laufen, darstellen.

Im Rahmen der Entwurfsraumanalyse gilt es, die optimale Systemgröße – also die Anzahl der IP-Cores und entsprechender Router zu ermitteln. Außerdem sollen die Bitbreite der Links und die Puffertiefe der Speicher an den Eingangs-Ports optimiert werden. Um dies zu erreichen, wurden die Kombinationen (siehe Tabelle 5.3) der Parameter in der Simulationsumgebung eingestellt und jeweils die drei Anwendungsbeispiele bzw. TGS abgearbeitet. Die verstrichene Zeit bei 1 GHz NoC-Takt geht in die Metrik ein. Als zweite relevante Größe wird die Leistungsaufnahme der Router aufgenommen. Anschließend berechnet sich das EDP aus den ermittelten Größen für jede Parameterkombination.

Für A1 ergeben sich für verschiedene Kombinationen der untersuchten Parameter unterschiedliche optimale Systemgrößen (siehe Abbildung 5.8). Mit 32 Bit Linkbreite und einer Pufferstufe arbeitet das System nach der gewählten Metrik mit 36 IP-Cores

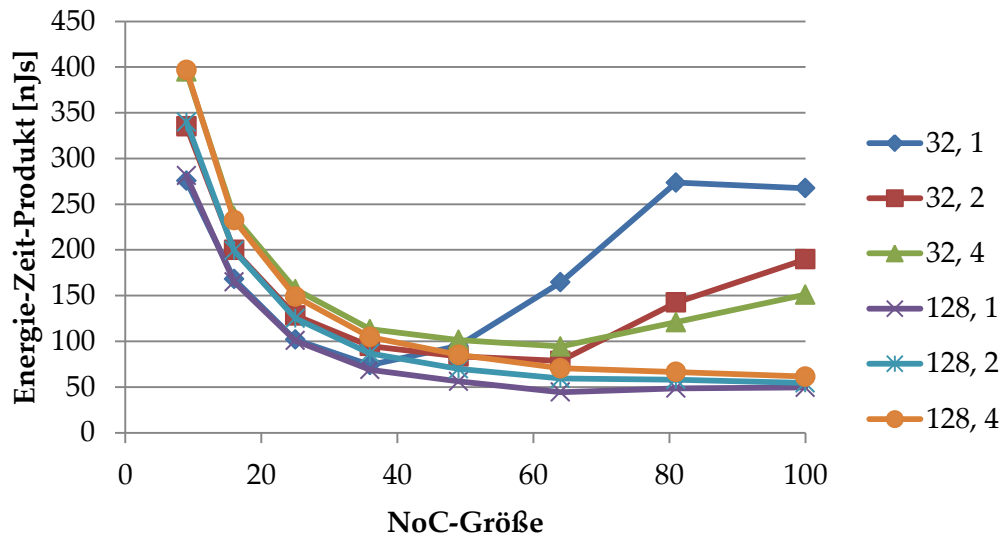


Abbildung 5.8: EDP für Applikation 1 bei Linkbreite 32 und 128 Bit sowie 1, 2 und 4 FIFO-Stufen

am effizientesten. Fügt man eine FIFO-Stufe hinzu, läuft die Anwendung beträchtlich schneller. Die Ausführungszeit beträgt 1,2 statt 1,8 ms. Die Leistungsaufnahme des NoC steigt jedoch, da mehr Register aktiv werden, sobald Flits an den Router-Ports anliegen. Diese Änderung der beiden Eingangsgrößen für das EDP führt zu einer Verschiebung der optimalen Systemgröße hin zu größeren NoCs. Das Optimum für 32 Bit und zwei sowie vier Pufferstufen liegt bei 64 IP-Cores. Wenn man die Bitbreite der Links auf 128 vergrößert, ergibt sich ein ähnliches Bild. Bei einer FIFO-Stufe ist eine Systemgröße von 64 IP-Cores optimal. Für zwei und vier Stufen verbessert sich die Metrik mit zunehmender NoC-Größe. Die Ergebnisse für 64 Bit Linkbreite liegen jeweils zwischen denen von 32 und 128 Bit. Im direkten Vergleich bietet die Kombination von 128 Bit und einer Pufferstufe für A1 das effizienteste System. In Tabelle 5.3 sind die zu jeder Kombination von Anwendungsszenario, Linkbreite und Puffertiefe optimalen NoC-Größen mit ihrem EDP aufgeführt.

Im Anwendungsbeispiel 2 gibt es weniger parallele Tasks, dafür nehmen diese eine größere Berechnungszeit in Anspruch und erzeugen mehr Daten. In der Folge liegt das Optimum der NoC-Größe niedriger (siehe Abbildung 5.9). Für das kleinste NoC mit 32 Bit Linkbreite und einer FIFO-Stufe besitzt das effizienteste System 25 IP-Cores. Die Version



Tabelle 5.3: Effizienz der Anwendungen unter Verwendung der verschiedenen Systemparameter

Anwendung	Linkbreite [Bit]	Puffertiefe	NoC-Größe	EDP [nJs]
A1	32	1	36	74,06
	32	2	64	78,48
	32	4	64	94,38
	64	1	64	49,93
	64	2	64	60,31
	64	4	64	74,48
	128	1	<b>64</b>	<b>44,50</b>
	128	2	100	54,45
	128	4	100	61,34
A2	32	1	25	142,27
	32	2	36	134,97
	32	4	36	157,30
	64	1	64	87,19
	64	2	64	82,16
	64	4	64	100,70
	128	1	64	55,97
	128	2	<b>64</b>	<b>54,78</b>
	128	4	81	72,01
A3	32	1	25	675,07
	32	2	36	640,68
	32	4	36	743,48
	64	1	36	374,92
	64	2	64	390,68
	64	4	64	427,68
	128	1	<b>64</b>	<b>237,44</b>
	128	2	100	255,10
	128	4	100	273,31

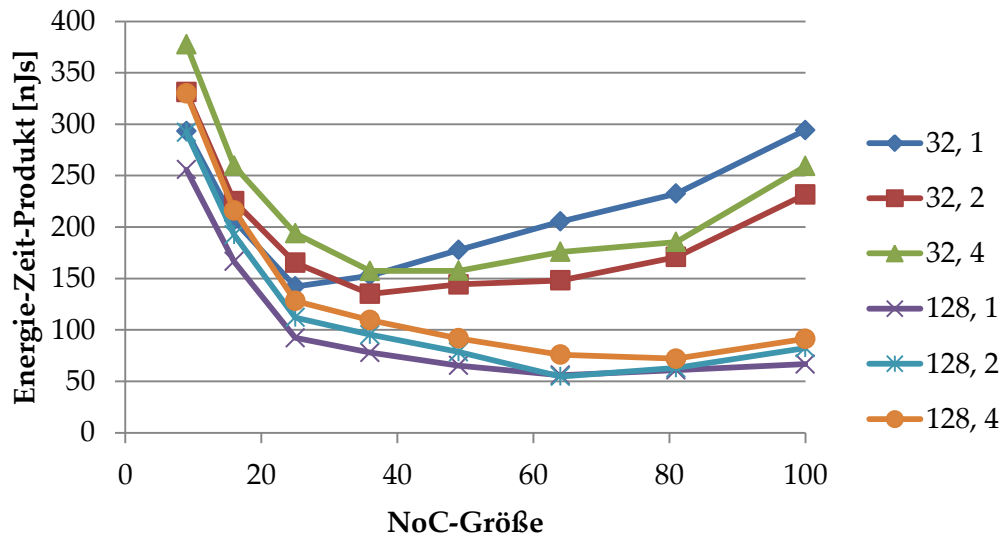


Abbildung 5.9: EDP für Applikation 2 bei Linkbreite 32 und 128 Bit sowie 1, 2 und 4 FIFO-Stufen

mit zwei Pufferstufen ist mit 36 IP-Cores etwas effizienter. Das Einfügen von vier FIFO-Stufen hebt das EDP wiederum etwas an, das Optimum liegt hier ebenfalls bei 36 IP-Cores. Die 64 Bit-Varianten sind etwas effizienter. Das Erweitern auf 128 Bit Linkbreite bringt auch für A2 die besten Ergebnisse. Ein System mit 64 IP-Cores und zwei Pufferstufen zeigt das geringste EDP.

Die Ergebnisse für das Anwendungsbeispiel 3 (siehe Abbildung 5.10) ähneln denen von A1. Die Systeme mit 32 Bit Linkbreite sind jedoch schon mit 25 bzw. 36 IP-Cores am effizientesten. Insgesamt wird auch hier das effizienteste EDP vom System mit 128 Bit Linkbreite, zwei FIFO-Stufen und 64 IP-Cores erreicht.

Die Ergebnisse zeigen, dass die Effizienz des NoC vor allem von der Systemgröße abhängt. Je mehr IP-Cores und NoC-Router vorhanden sind, desto mehr kann die erhöhte NoC-Leistungsfähigkeit durch größere Linkbreiten und mehr FIFO-Speicher ausgespielt werden.

Ein weiteres Ergebnis der Systemsimulation ist die Temperaturverteilung, die mit dem in Kapitel 5.6 beschriebenen Modell erstellt wird. In Abbildung 5.11 ist ein beispielhafter zeitlicher sowie örtlicher Verlauf dargestellt. Das System beginnt zunächst in einem Grundzustand, in dem moderate Aktivität angenommen wird. Danach erzeugt ein Task

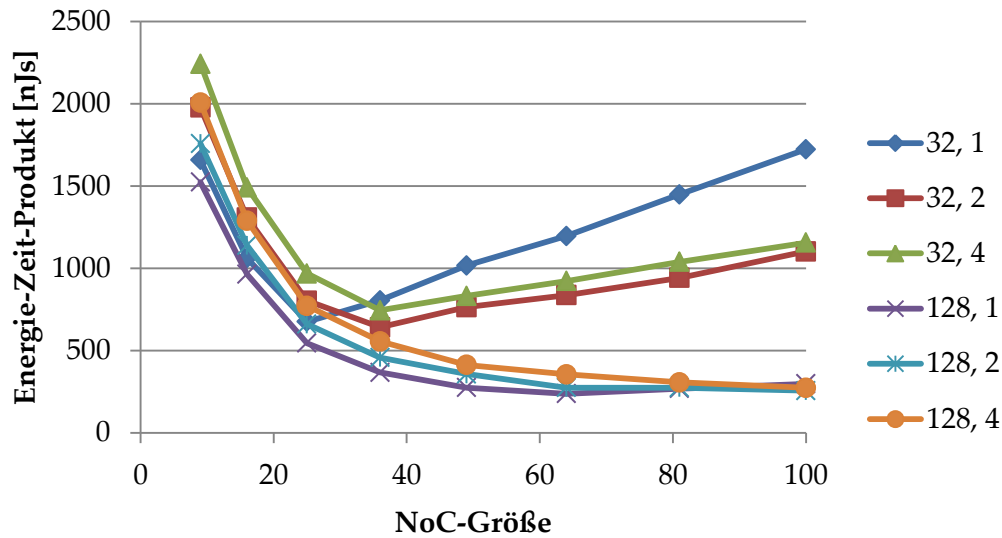


Abbildung 5.10: EDP für Applikation 3 bei Linkbreite 32 und 128 Bit sowie 1, 2 und 4 FIFO-Stufen

auf dem IP-Core unten links eine große Last (siehe Abbildung 5.11(b)). In diesem Fall wurde der Temperaturverlauf mit einer Auflösung simuliert, die einem Block pro Link, Router sowie IP-Core entspricht. Die in der Abbildung erscheinenden Verläufe entstehen durch Interpolation. Die an den aktiven Bereich angrenzenden Router und Links werden dabei sowohl durch passive Übertragung der Wärme aus dem IP-Core, als auch durch ihre eigene gesteigerte Aktivität über die anfangs moderate Temperatur hinaus erwärmt. Die sich über die Zeit auf die benachbarten Elemente ausbreitende Wärme ist gut zu erkennen. Auf diese Weise können drohende Hotspots detektiert werden und Design-Parameter angepasst werden. Weiterhin kann ein temperaturgesteuertes Laufzeitmanagement simuliert werden, mit dem eine Steuerung des Systems über DVFS oder das Verschieben von Tasks in kühlere Bereiche möglich wird [WCTT11, WGTU11].

## 5.8 Fazit

In diesem Kapitel wurde zunächst ein Systemsimulator vorgestellt, der die schnelle Evaluation der Leistungsfähigkeit eines NoC ermöglicht und dabei zyklenakkurate Ergebnisse liefert. Der Geschwindigkeitsvorteil gegenüber einer RTL-Implementierung liegt – abhän-

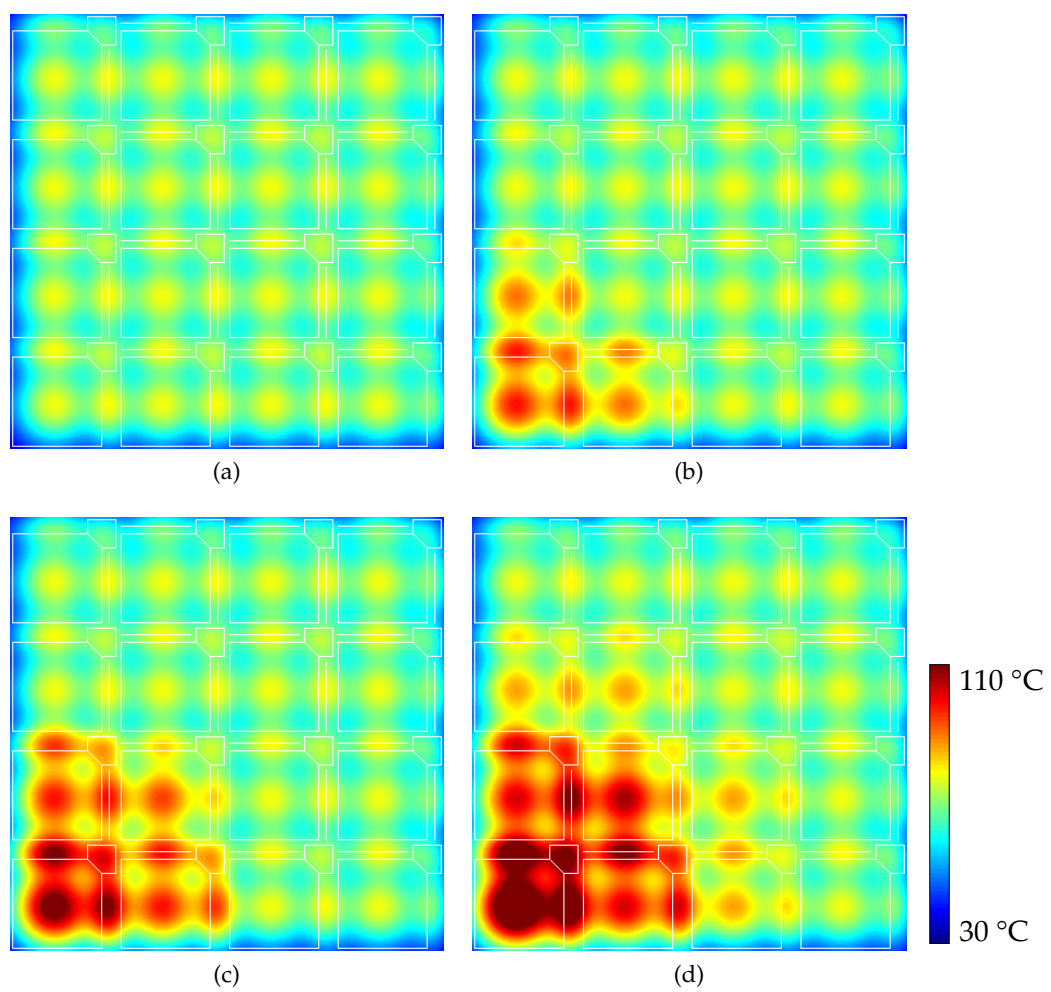


Abbildung 5.11: Temperaturverlauf

gig von Betriebsparametern, wie Systemgröße und Auslastung – im Bereich von 15- bis 80-fach. Diese Simulationsumgebung wurde um die Rückannotation der Leistungsaufnahme aus den vorangegangenen Untersuchungen auf Gatterebene ergänzt. Außerdem wurde ein Temperaturmodell implementiert, das die Ausbreitung der Wärme auf dem Chip nachstellt. Um den Datenverkehr von verschiedenen Applikationen imitieren zu können, wurden Task-Graphen implementiert. Diese können mit verschiedenen Parametern erzeugt werden, was zu unterschiedlicher Parallelität führt. Außerdem kann die Anzahl und Größe der Pakete beeinflusst werden. Für die Durchführung der weiteren Versuche wurde eine Task-Mapping-Strategie implementiert, die die Parallelität der Task-Graphen möglichst optimal auf das vorliegende System umsetzt.

Um das erstellte System zu testen, wurden verschiedene Task-Graph-Sets generiert, die bestimmten Arten von Anwendungen entsprechen. Einerseits wurde die Parallelität der einzelnen Tasks, andererseits die Anzahl und Größe der Pakete verändert. Mit Hilfe der erstellten Simulationsumgebung konnten verschiedene Parameter des NoC – wie Systemgröße, Puffertiefe und Linkbreite – getestet werden. Die Ergebnisse wurden vor allem hinsichtlich des Power-Delay-Produktes analysiert. Es ergaben sich verschiedene Parameterkombinationen für die unterschiedlichen Anwendungen, die nach der angewandten Metrik als optimal gelten. Auf diese Weise lässt sich beispielsweise für eine gegebene Anwendung das optimale NoC finden oder für ein gegebenes NoC Betriebsdaten für verschiedene Anwendungen ermitteln.



# Kapitel 6

## Zusammenfassung

### 6.1 Ergebnisse der Arbeit

Die rasante technologische Entwicklung, die die Steigerung der Transistordichte auf Mikrochips möglich macht, hat eine Zunahme der Leistungsfähigkeit zur Folge. In dieser Arbeit wurden zunächst die fertigungstechnologischen Fortschritte und deren unmittelbare Folgen betrachtet. Es wurde vor allem auf bestehende Probleme bezüglich der Leistungsaufnahme eines Systems als auch deren Dichte eingegangen. Problematische Folgen wurden hauptsächlich im thermalen Bereich identifiziert. Entstehende Hotspots oder extreme Temperaturwechsel können die Zuverlässigkeit beeinträchtigen. Weiterhin beschleunigt die dauerhafte Erhöhung der Temperatur exponentiell verschiedene Alterungseffekte, die die Performance und letztendlich ebenfalls die Zuverlässigkeit mikroelektronischer Systeme gefährden.

Das effiziente Ausnutzen der technischen Möglichkeiten stellt während der Entwicklung eine weitere Herausforderung dar. Dieses Problem wird mit verschiedenen Ansätzen abgemildert. Neben der zunehmenden Automatisierung der Entwurfsschritte ist eine zeitliche Parallelisierung des Hard- und Softwareentwurfs nötig. Das Hardwaresystem muss deshalb schon in einem frühen Entwurfsstadium simulationsfähig sein und möglichst exakte Aussagen über verschiedene Parameter – wie Performance und Leistungsaufnahme – ermöglichen.

Eine weitere Reaktion auf die technologischen Weiterentwicklungen ist die Modularisierung der Systeme. Dadurch werden bestimmte Anforderungen an die Kommunikati-

onssysteme gestellt, die durch die Nutzung von NoCs sehr gut erfüllt werden können. NoCs eignen sich vor allem für Systeme in denen die Wiederverwendung von Einzelkomponenten und eine große Skalierbarkeit wichtige Faktoren sind. Nachdem NoCs in der wissenschaftlichen Betrachtung schon seit einigen Jahren intensiv untersucht werden, sind in den letzten zwei Jahren auch die ersten kommerziellen Produkte erschienen, die sich des Konzeptes bedienen.

Nach diesen grundlegenden Betrachtungen ist ein Bedarf an unterstützenden Entwurfswerkzeugen zu identifizieren, die eine Systemkomposition und das Festlegen bestimmter Parameter des Kommunikationssystems möglich machen. Dabei muss darauf geachtet werden, dass eine frühe und effiziente Systemsimulation möglich sein soll. Die Anwendung eines solchen Werkzeugs soll möglichst früh im Entwurfsablauf ausreichend genaue Aussagen zu Leistungsumsatz und Temperaturverteilung innerhalb des Systems sowie eine hohe Auflösung der einstellbaren Parameter ermöglichen.

Um diese Aufgaben zu erfüllen, wurden einerseits die Übertragungskanäle (Links) und andererseits die Router als Bestandteile des NoC betrachtet. Neben der Bestimmung der Leistungsaufnahme von Links wurde besonders auf den Crosstalk-Effekt auf längeren Leitungen eingegangen. Dieses Phänomen kann Einfluss auf die Signalqualität, die Übertragungszeit und den Leistungsumsatz haben. Die Verzögerung der Signale wurde in einem praktischen Versuchsaufbau mit Hilfe von FPGAs untersucht. Die erzielten Messungen entsprechen qualitativ den theoretischen Annahmen. Obwohl eine Überprüfung der absoluten Werte aufgrund der unbekannten Koppelkapazitäten nicht möglich ist, zeigt sich in diesem Versuch, wie einflussreich der Crosstalk-Effekt selbst auf programmierbaren Schaltungen sein kann. Der Einfluss von Crosstalk auf die benötigte Übertragungsenergie hängt im Wesentlichen von der Häufigkeit der unterschiedlichen Crosstalk-Muster ab. Eine Untersuchung von synthetischen und realen Datenmustern zeigt, dass Energieaufwand und Leistungsumsatz im Durchschnitt nicht vom Crosstalk-Effekt beeinflusst werden, da sich die Nach- und Vorteile des Crosstalk-Effektes im Mittel ausgleichen. Die Erzeugung des Worst-Case-Datenmusters zeigt jedoch, dass eine Berücksichtigung von Crosstalk für die Bestimmung der maximalen Leistungsaufnahme unabdingbar ist.

Weiterhin wurde ein parametrisierbarer Router-Entwurf mit Optimierungen wie Clock-Gating-Mechanismen ausgestattet und eingehend auf seine Eigenschaften untersucht. Dazu gehören Ergebnisse zur benötigten Fläche, der erreichbaren Betriebsfrequenz und der Leistungsaufnahme. Die Performance des Routers kann mit den Parametern Linkbrei-



te und Puffertiefe beeinflusst werden. Alle Untersuchungen wurden für drei verschiedene Fertigungstechnologien durchgeführt. Als Ergebnis wird eine Annotationsmethode entwickelt, die die Leistungsaufnahme für die konfigurierbaren Parameter durch Interpolation bestimmt. Dadurch müssen nur sehr wenige Synthese- und Simulationsvorgänge auf Gatterebene ablaufen. Dieses Vorgehen ermöglicht eine effiziente Simulation auf Systemebene und verkürzt damit die Entwurfszeit.

Schließlich wird eine dem vorgeschlagenen Konzept entsprechende Erweiterung der Simulationsumgebung vorgestellt, die die Simulation eines heterogenen NoC-basierten Many-Core-Systems ermöglicht. Das mit Hilfe von SystemC implementierte Werkzeug kann synthetisch erzeugte Task-Graphen nutzen, um verschiedene Systemauslastungen zu simulieren. Außerdem können auf diese Weise verschiedene Anwendungsszenarien als Arbeitslast für das System erstellt werden, die sich beispielsweise in ihrer Parallelität und der Menge der benötigten sowie erzeugten Daten unterscheiden. Die Simulationsgeschwindigkeit der SystemC-Lösung ist bis zu 80-mal höher als eine vergleichbare Implementierung auf RT-Ebene. Letztendlich wird eine Aussage zur optimalen Dimensionierung der verschiedenen NoC-Parameter ermöglicht, um für gegebene Anwendungen eine hohe Effizienz bezüglich Leistungsbedarf und Performance zu ermöglichen. Dies wird an verschiedenen Task-Graphen mit der Zielstellung eines niedrigen Power-Delay-Produktes gezeigt. Hier zeigt sich, dass größere Systeme ein zusätzliches Kommunikationsaufkommen erzeugen. Dies führt zur Zunahme der Leistungsaufnahme, welcher die potentielle Verkürzung der benötigten Verarbeitungszeit gegenübersteht. Ab einer bestimmten Systemgröße ist die weitere Parallelisierung der Anwendung jedoch nicht mehr effizient genug. Das so ermittelte optimale System für ein bestimmtes Anwendungsszenario stellt somit einen Kompromiss aus Verarbeitungszeit und Leistungsaufnahme dar.

Zusätzlich wurde eine Möglichkeit geschaffen, mit den in der Simulation erzeugten Daten eine Vorhersage der Temperaturverteilung innerhalb des Chips zu ermöglichen. Damit lässt sich der vorgestellte Simulator um temperaturorientierte Laufzeitverfahren erweitern. Dazu zählen z. B. das Verschieben von Tasks sowie dynamische Frequenz- und Spannungsanpassungen.

## 6.2 Kritik und Ausblick

Die hier vorgestellte Methode, den zeitlichen Crosstalk-Effekt innerhalb einer Schaltung zu messen, wurde bisher nur auf zwei FPGA-Typen angewandt. In zukünftigen Untersuchungen sollten weitere programmierbare Schaltungen mit Hilfe der Messmethode überprüft werden. Vor allem bei neueren Modellen in modernsten Fertigungstechnologien besteht eine erhöhte Anfälligkeit für Crosstalk. Ob diese Effekte in praktischen FPGA-Entwürfen eine Rolle spielen, konnte in dieser Arbeit nicht nachgewiesen werden. Dies und inwiefern die verfügbaren Routing-Tools anfällige Leitungsführungen erkennen, bedarf weiterer Untersuchungen.

Das in dieser Arbeit vorgestellte Vorgehen, die Leistungsaufnahme und Temperaturverteilung eines mikroelektronischen Kommunikationssystems zu ermitteln, kann bereits früh im Entwurfsprozess angewandt werden. Die Ermittlung des Leistungsbedarfs basiert auf Rückannotationen. Es wurde Wert darauf gelegt, dass die Abschätzung auf relevanten Simulationsdaten basieren kann, ohne die Simulationsgeschwindigkeit zu stark zu beeinflussen. Im Rahmen der durchgeführten Rückannotationen der Leistungswerte setzen sich kleine Fehler und Ungenauigkeiten, die beispielsweise bei der Charakterisierung auf Zellebene auftreten, auf der Systemebene fort. Dies kann eine Addition der absoluten Fehler bedeuten. Die Relationen zwischen den Entwurfsalternativen und die relative Abweichung steigen durch die angewandten Techniken jedoch nicht. Damit lassen sich qualitative Aussagen zum Vergleich der untersuchten Systemkonfigurationen treffen. Um belastbare quantitative Ergebnisse zu erhalten, muss zusätzlicher Aufwand in die Verifikation der verwendeten Leistungswerte investiert werden. Dies könnte z. B. durch die Implementierung des Gesamtsystems bis auf mindestens die Gatterebene erreicht werden.

Die vorgestellten Methoden bringen nur dann Vorteile, wenn ähnliche Grundstrukturen für verschiedene Topologien und Ausgestaltungen des Kommunikationssystems genutzt werden sollen. Andernfalls müssen viele Synthesevorgänge für konkrete Technologien durchgeführt werden. Dann würde die gesamte Methodik keinen besonderen Zeitvorteil mehr mit sich bringen. Alternativ würde es sich in einem solchen Fall anbieten die grundlegenden Annotationen nicht auf Basis von Syntheseergebnissen sondern anderen analytischen Abschätzungen durchzuführen [ZWZG12, KLPS09]. Dies führt zu größeren Abweichungen aber einer erhöhten Flexibilität. Weiterhin ist die beschriebene Methode

technologieabhängig, was zu ähnlichen Nachteilen führt, sollte eine große Anzahl an möglichen Zieltechnologien zur Verfügung stehen.

Da die Annahmen zur Energieaufnahme der Links auf einem Floorplan basieren, sind bereits zum Zeitpunkt der Systemsimulation und Entwurfsraumanalyse Angaben zum grundsätzlichen Layout der späteren Schaltung nötig. Da diese Informationen zu einem solch frühen Zeitpunkt im Entwicklungsfluss nur eingeschränkt zur Verfügung stehen, können an dieser Stelle ebenfalls Fehler und Ungenauigkeiten in die Abschätzung eingebracht werden.

Weiterhin bestehen Erweiterungsmöglichkeiten in der Einbeziehung der Performance- und Flächenergebnisse aus der Synthese in die Systemsimulation analog zur in dieser Arbeit vorgestellten Vorgehensweise. Mit diesem Schritt würde die Aussagekraft der Analyse auf Systemebene um ein weitere Entwurfskriterien ergänzt.



# Literaturverzeichnis

- [Ada13] ADAPTEVA: <http://www.adapteva.com/>. Abruf am 11.10.2013 (zitiert auf S. 34 und 65)
- [AKPJ09] AGARWAL, Niket ; KRISHNA, Tushar ; PEH, Li-Shiuan ; JHA, Niraj K.: GARNET: A detailed on-chip network model inside a full-system simulator. In: *IEEE International Symposium on Performance Analysis of Systems and Software IEEE*, 2009 (ISPASS), S. 33–42 (zitiert auf S. 92)
- [Amd67] AMDAHL, Gene M.: Validity of the single processor approach to achieving large scale computing capabilities. In: *Proceedings of the AFIPS '67 Spring Joint Computer Conference*. New York, NY, USA : ACM, 1967, S. 483–485 (zitiert auf S. 107)
- [BCGK04] BOLOTIN, Evgeny ; CIDON, Israel ; GINOSAR, Ran ; KOLODNY, Avinoam: QNoC: QoS architecture and design process for network on chip. In: *Journal of Systems Architecture* 50 (2004), Nr. 2 - 3, S. 105 – 128. – Special issue on networks on chip (zitiert auf S. 33)
- [BDM02] BENINI, Luca ; DE MICHELI, Giovanni: Networks on chip: a new paradigm for systems on chip design. In: *Proceedings of Design, Automation and Test in Europe, 2002 (DATE)*, S. 418–419 (zitiert auf S. 31)
- [Bla69] BLACK, James R.: Electromigration failure modes in aluminum metallization for semiconductor devices. In: *Proceedings of the IEEE* 57 (1969), Nr. 9, S. 1587–1594 (zitiert auf S. 22)
- [BM02] BANERJEE, Kaustav ; MEHROTRA, Amit: A power-optimal repeater insertion methodology for global interconnects in nanometer designs. In: *IEEE Transactions on Electron Devices* 49 (2002), Nr. 11, S. 2001–2007 (zitiert auf S. 40)
- [BMKC05] BHOJWANI, Praveen ; MAHAPATRA, Rabi ; KIM, Eun J. ; CHEN, Thomas: A Heuristic for Peak Power Constrained Design of Network-on-Chip (NoC) Based Multimode Systems. In: *International Conference on VLSI Design* (2005), S. 124–129 (zitiert auf S. 93)
- [BPP<sup>+</sup>11] BEKIARIS, Dimitris ; PAPANIKOLAOU, Antonis ; PAPAMELETIS, Christos ; SOUDRIS, Dimitrios ; ECONOMAKOS, George ; PEKMESTZI, Kiamal: A temperature-aware time-dependent dielectric breakdown analysis framework. In: *Proceedings of the 20th International Conference on Integrated Circuit and System Design: Power and Timing modeling, Optimization and Simulation*. Berlin, Heidelberg : Springer-Verlag, 2011 (PATMOS'10), S. 73–83 (zitiert auf S. 21)

- [BTM00] BROOKS, David ; TIWARI, Vivek ; MARTONOSI, Margaret: Wattch: A framework for architectural-level power analysis and optimizations. In: *SIGARCH Computer Architecture News* 28 (2000), Nr. 2, S. 83–94 (zitiert auf S. 84)
- [CGLP04] CURABA, Stephane ; GRAMMATIKAKIS, Miltos D. ; LOCATELLI, Riccardo ; PAPARIELLO, Francesco: OCCN: a NoC modeling framework for design exploration. In: *Journal of Systems Architecture* (2004) (zitiert auf S. 93)
- [CGRB06] CIORDAS, Calin ; GOOSSENS, Kees ; RADULESCU, Andrei ; BASTEN, Twan: NoC monitoring: impact on the design flow. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2006 (ISCAS), S. 1981–1984 (zitiert auf S. 65)
- [CHE11] CARLSON, T.E. ; HEIRMAN, W. ; EECKHOUT, L.: Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In: *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2011, S. 1–12 (zitiert auf S. 93)
- [CKK00] CHEN, Pinhong ; KIRKPATRICK, Desmond A. ; KEUTZER, Kurt: Miller Factor for Gate-level Coupling Delay Calculation. In: *Proceedings of the IEEE/ACM International Conference on Computer-aided Design*, 2000 (ICCAD), S. 68–75 (zitiert auf S. 42)
- [Cor09] CORNELIUS, Claas: *Design of complex integrated systems based on networks-on-chip - Trading off performance, power and reliability*, Universität Rostock, Diss., 2009 (zitiert auf S. 24, 40 und 69)
- [CPU13] CPU WORLD: <http://www.cpu-world.com>. Abruf am 11.06.2013 (zitiert auf S. 14)
- [DGY<sup>+</sup>08] DAI, Mingzhi ; GAO, Chao ; YAP, Kinleong ; SHAN, Yi ; CAO, Zigui ; LIAO, Kuan-gyang ; WANG, Liang ; CHENG, Bo ; LIU, Shaohua: A model with temperature-dependent exponent for hot-carrier injection in high-voltage nMOSFETs involving hot-hole injection and dispersion. In: *IEEE Transactions on Electron Devices* 55 (2008), Nr. 5, S. 1255–1258 (zitiert auf S. 21)
- [DLK10] DUAN, Chunjie ; LAMERES, Brock J. ; KHATRI, Sunil P.: *On and Off-Chip Crosstalk Avoidance in VLSI Design*. 1st. Springer Publishing Company, Incorporated, 2010. – ISBN 144190946X, 9781441909466 (zitiert auf S. 58)
- [DRR11] DAS, Nachiketa ; ROY, Pranab ; RAHAMAN, Hafizur: Runtime Congestion and Crosstalk Aware Router for FPGA Using Jbits3.0 for Partial Reconfigurable Application. In: *International Symposium on Electronic System Design*, 2011 (ISED), S. 146–151 (zitiert auf S. 59)
- [DRW98] DICK, Robert P. ; RHODES, David L. ; WOLF, Wayne: TGFF: task graphs for free. In: *Proceedings of the Sixth International Workshop on Hardware/Software Codesign. (CODES/CASHE)*, 1998, S. 97–101 (zitiert auf S. 99)
- [EBSA<sup>+</sup>11] ESMAEILZADEH, Hadi ; BLEM, Emily ; ST. AMANT, Renee ; SANKARALINGAM, Karthikeyan ; BURGER, Doug: Dark silicon and the end of multicore scaling. In:

- Proceedings of the International Symposium on Computer Architecture*. New York, NY, USA : ACM, 2011 (ISCA), S. 365–376 (zitiert auf S. 2 und 15)
- [FLD<sup>+</sup>08] FLEMING, Kermin ; LIN, Chun-Chieh ; DAVE, Nirav ; ARVIND ; RAGHAVAN, Gopal ; HICKS, Jamey: H.264 Decoder: A case study in multiple design points. In: *6th ACM/IEEE International Conference on Formal Methods and Models for Co-Design*, 2008 (MEMOCODE), S. 165–174 (zitiert auf S. 54)
- [GAA<sup>+</sup>00] GOLDBLATT, R. D. ; AGARWALA, B. ; ANAND, M. B. ; BARTH, E. P. u. a.: A high performance 0.13  $\mu\text{m}$  copper BEOL technology with low-k dielectric. In: *Proceedings of the IEEE International Interconnect Technology Conference*, 2000 (IITC), S. 261–263 (zitiert auf S. 39)
- [GCTK13] GORSKI, Philipp ; CORNELIUS, Claas ; TIMMERMANN, Dirk ; KÜHN, Volker: Red-NoCs: A runtime configurable solution for cluster-based and multi-objective system management in Networks-on-Chip. In: *The Eighth International Conference on Systems*, 2013 (ICONS), S. 192–201 (zitiert auf S. 65)
- [GDR05] GOOSSENS, Kees ; DIELISSSEN, John ; RADULESCU, Andrei: AEthereal network on chip: concepts, architectures, and implementations. In: *Design & Test of Computers*, IEEE 22 (2005), Nr. 5, S. 414–421 (zitiert auf S. 33)
- [GGWT11] GAG, Martin ; GORSKI, Philipp ; WEGNER, Tim ; TIMMERMANN, Dirk: Evaluation of Switch-to-Switch Header Flit Protection Schemes in Networks-on-Chip. In: *GMM/GI/ITG- Fachtagung Zuverlässigkeit und Entwurf*, 2011 (ZuE), S. 108–115 (zitiert auf S. 76)
- [Ghe06] GHENASSIA, Frank: *Transaction-Level Modeling with SystemC: TLM Concepts and Applications for Embedded Systems*. Springer, 2006 (zitiert auf S. 27)
- [GJB<sup>+</sup>10] GANESAN, Karthik ; JO, Jungho ; BIRCHER, W. L. ; KASERIDIS, Dimitris ; YU, Zhibin ; JOHN, Lizy K.: System-level Max Power (SYMPO): A Systematic Approach for Escalating System-level Power Consumption Using Synthetic Benchmarks. In: *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*. New York, NY, USA : ACM, 2010 (PACT), S. 19–28 (zitiert auf S. 57)
- [GN92] GLASS, Christopher J. ; NI, Lionel M.: The turn model for adaptive routing. In: *Proceedings of the 19th International Symposium on Computer Architecture*. New York, NY, USA : ACM, 1992 (ISCA), S. 278–287 (zitiert auf S. 62)
- [GPB09] GANGULY, Amlan ; PANDE, Partha P. ; BELZER, Benjamin: Crosstalk-aware channel coding schemes for energy efficient and reliable NOC interconnects. In: *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on 17 (2009), Nr. 11, S. 1626–1639 (zitiert auf S. 58)
- [GWG<sup>+</sup>13] GAG, Martin ; WEGNER, Tim ; GORSKI, Philipp ; TOCKHORN, Andreas ; TIMMERMANN, Dirk: System level modeling of Networks-on-Chip for power estimation and design space exploration. In: *16. Workshop Methoden und Beschreibungssprachen zur*

- Modellierung und Verifikation von Schaltungen und Systemen*, 2013 (MBMV), S. 25–34 (zitiert auf S. 61, 91, 92 und 101)
- [GWT10] GAG, Martin ; WEGNER, Tim ; TIMMERMAN, Dirk: System level power estimation of System-on-Chip interconnects in consideration of transition activity and crosstalk. In: *International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2010 (PATMOS), S. 21–30 (zitiert auf S. 37)
- [GWWT12] GAG, Martin ; WEGNER, Tim ; WASCHKI, Ansgar ; TIMMERMAN, Dirk: Temperature and on-chip crosstalk measurement using ring oscillators in FPGA. In: *15th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2012 (DDECS), S. 201–204 (zitiert auf S. 37)
- [HAAN<sup>+</sup>07] HOSSAIN, Hemayet ; AHMED, Mostak ; AL-NAYEEM, Abdullah ; ISLAM, Tanzima Z. ; AKBAR, Mostofa: Gpnocsim - A General Purpose Simulator for Network-On-Chip. In: *International Conference on Information and Communication Technology*, 2007 (ICICT), S. 254–257 (zitiert auf S. 92)
- [HBB<sup>+</sup>11] HENKEL, Jörg ; BAUER, Lars ; BECKER, Joachim ; BRINGMANN, Oliver ; BRINKSCHULTE, Uwe ; CHAKRABORTY, Samarjit ; ENGEL, Michael ; ERNST, Rolf ; HÄRTIG, Hermann ; HEDRICH, Lars ; HERKERSDORF, Andreas ; KAPITZA, Rüdiger ; LOHMANN, Daniel ; MARWEDEL, Peter ; PLATZNER, Marco ; ROSENSTIEL, Wolfgang ; SCHLICHTMANN, Ulf ; SPINCZYK, Olaf ; TAHOORI, Mehdi ; TEICH, Jürgen ; WEHN, Norbert ; WUNDERLICH, Hans-Joachim: Design and architectures for dependable embedded systems. In: *Proceedings of the Seventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. New York, NY, USA : ACM, 2011 (CODES+ISSS), S. 69–78 (zitiert auf S. 17 und 18)
- [HDV<sup>+</sup>11] HOWARD, Jason ; DIGHE, Saurabh ; VANGAL, Sriram R. ; RUHL, Gregory ; BORKAR, Nitin ; JAIN, Shailendra ; ERRAGUNTALA, Vasanth ; KONOW, Michael ; RIEPEN, Michael ; GRIES, Matthias ; DROEGE, Guido ; LUND-LARSEN, Tor ; STEIBL, Sebastian ; BORKAR, Shekhar ; DE, Vivek K. ; VAN DER WIJNGAART, Rob: A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. In: *IEEE Journal of Solid-State Circuits* 46 (2011), Nr. 1, S. 173–183 (zitiert auf S. 34)
- [HM04] HU, Jingcao ; MARCULESCU, Radu: DyAD: Smart Routing for Networks-on-chip. In: *Proceedings of the 41st Annual Design Automation Conference*. New York, NY, USA : ACM, 2004 (DAC). – ISBN 1–58113–828–8, S. 260–263 (zitiert auf S. 92)
- [IEE90] IEEE: IEEE Standard Glossary of Software Engineering Terminology. In: *IEEE Std 610.12-1990* (1990) (zitiert auf S. 17)
- [IEE12] IEEE: IEEE Standard SystemC Language Reference Manual. In: *IEEE Std 1666-2011* (2012) (zitiert auf S. 94)
- [Int04] INTEL: *White Paper 30057701. Wireless intel speedstep power manager: Optimizing power consumption for the intel pxa27x processor family*. 2004 (zitiert auf S. 15)



- [ITR12] ITRS: The International Technology Roadmap for Semiconductors. (2012) (zitiert auf S. 8, 16, 25 und 26)
- [ITU13] ITU INTERNATIONAL TELECOMMUNICATION UNION: *Latest global technology development figures*. press release, 2013 (zitiert auf S. 1)
- [J<sup>+</sup>00] JANTSCH, Axel u. a.: *NOSTRUM*. Department for Electronics, Computer and Software Systems, KTH, Stockholm, Schweden, 2000 (zitiert auf S. 33 und 93)
- [JAHG<sup>+</sup>07] JAIN, Lavina ; AL-HASHIMI, BM ; GAUR, MS ; LAXMI, V ; NARAYANAN, A: NIRGAM: a simulator for NoC interconnect routing and application modeling. In: *Design, Automation and Test in Europe Conference*, 2007 (zitiert auf S. 93)
- [JEDEC10] JOINT ELECTRON DEVICE ENGINEERING COUNCIL, JEDEC: *Failure Mechanisms and Models for Silicon Semiconductor Devices*. 2010 (EIA/JEDEC publication) (zitiert auf S. 20, 21 und 23)
- [JS08] JOOST, Ralf ; SALOMON, Ralf: BOUNCE, a concept to measure picosecond time intervals with standard hardware. In: *IEEE International Conference on Emerging Technologies and Factory Automation*, 2008 (ETFA), S. 1010–1015 (zitiert auf S. 45)
- [Kal14] KALRAY: <http://www.kalray.eu/>. Abruf am 07.01.2014 (zitiert auf S. 33)
- [KLPS09] KAHNG, Andrew B. ; LI, Bin ; PEH, Li-Shiuan ; SAMADI, Kambiz: ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration. In: *Proceedings of the Conference on Design, Automation and Test in Europe*, 2009 (DATE), S. 423–428 (zitiert auf S. 40, 52, 84 und 122)
- [Kub11] KUBISCH, Stephan: *Architekturen fuer Ethernet-basierte Teilnehmerzugangsnetzwerke und deren Umsetzung in Hardware*, Universität Rostock, Diss., 2011 (zitiert auf S. 66 und 91)
- [LAS<sup>+</sup>09] LI, Sheng ; AHN, Jung H. ; STRONG, Richard D. ; BROCKMAN, Jay B. ; TULLSEN, Dean M. ; JOUPPI, Norman P.: McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In: *Annual IEEE/ACM International Symposium on Microarchitecture*, 2009 (MICRO), S. 469–480 (zitiert auf S. 84)
- [LAS<sup>+</sup>13] LI, Sheng ; AHN, Jung H. ; STRONG, Richard D. ; BROCKMAN, Jay B. ; TULLSEN, Dean M. ; JOUPPI, Norman P.: The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing. In: *ACM Transactions on Architecture and Code Optimization (TACO)* 10 (2013), Nr. 1, S. 5:1–5:29 (zitiert auf S. 40 und 84)
- [LKY<sup>+</sup>08] LEE, Kyong-Taek ; KANG, Chang-Yong ; YOO, Ook S. ; CHOI, Rino ; LEE, Byoung-Hun ; LEE, Jack C. ; LEE, Hi-Deok ; JEONG, Yoon-Ha: PBTI-associated high-temperature hot carrier degradation of nMOSFETs with metal-gate/high-k dielectrics. In: *Electron Device Letters, IEEE* 29 (2008), Nr. 4, S. 389–391 (zitiert auf S. 22)

- [LRC<sup>+</sup>11] LIS, Mieszko ; REN, Pengju ; CHO, Myong H. ; SHIM, Keun S. ; FLETCHER, Christopher W. ; KHAN, Omer ; DEVADAS, Srinivas: Scalable, Accurate Multicore Simulation in the 1000-core Era. In: *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, 2011 (ISPASS). – ISBN 978-1-61284-367-4, S. 175–185 (zitiert auf S. 92)
- [LZKC02] LU, Ruibing ; ZHONG, Guoan ; KOH, Cheng-Kok ; CHAO, Kai-Yuan: Flip-Flop and Repeater Insertion for Early Interconnect Planning. In: *Proceedings of the Conference on Design, Automation and Test in Europe*, 2002 (DATE), S. 690 (zitiert auf S. 40)
- [MG89] MEIJS, Nick P. d. ; GENDEREN, Arjan J.: An efficient finite element method for submicron IC capacitance extraction. In: *Proceedings of the 26th ACM/IEEE Design Automation Conference*. New York, NY, USA : ACM, 1989 (DAC), S. 678–681 (zitiert auf S. 39)
- [MKK<sup>+</sup>10] MILLER, Jason E. ; KASTURE, Harshad ; KURIAN, George ; GRUENWALD, Charles ; BECKMANN, Nathan ; CELIO, Christopher ; EASTEP, Jonathan ; AGARWAL, Anant: Graphite: A distributed parallel simulator for multicores. In: *IEEE International Symposium on High Performance Computer Architecture*, 2010 (HPCA), S. 1–12 (zitiert auf S. 93)
- [MM09] MOSCIBRODA, Thomas ; MUTLU, Onur: A case for bufferless routing in on-chip networks. In: *Proceedings of the 36th International Symposium on Computer Architecture*. New York, NY, USA : ACM, 2009 (ISCA), S. 196–207 (zitiert auf S. 65)
- [Moo65] MOORE, Gordon E.: Cramming more components onto integrated circuits. In: *Electronics Bd.* 38, 1965, S. 114–116 (zitiert auf S. 1 und 7)
- [MPR01] METRA, Cecilia ; PAGANO, Aandrea ; RICCO, Bruno: On-line testing of transient and crosstalk faults affecting interconnections of FPGA-implemented systems. In: *Proceedings on the IEEE International Test Conference*, 2001, S. 939–947 (zitiert auf S. 45)
- [MW97] MCNAUGHT, Alan D. ; WILKINSON, Andrew: *IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book")*. WileyBlackwell; 2nd Revised edition edition, 1997 (zitiert auf S. 19)
- [NC06] NARENDRA, Siva G. ; CHANDRAKASAN, Anantha P.: *Leakage in Nanometer CMOS Technologies*. Springer, 2006 (Series on integrated circuits and systems) (zitiert auf S. 10 und 11)
- [NVK<sup>+</sup>07] NAJEEB, K. ; VARDHAN, Vishnu Vardhan R. ; KONDA, R. ; KUMAR, Siva ; HARI, Sastry ; KAMAKOTI, V. ; VEDULA, Vivekananda M.: Power Virus Generation Using Behavioral Models of Circuits. In: *25th IEEE VLSI Test Symposium*, 2007, S. 35–42 (zitiert auf S. 57)
- [Par11] PARK, Jemin: *Improving CMOS Speed and Switching Power with Air-gap Structures*, EECS Department, University of California, Berkeley, Diss., Jul 2011 (zitiert auf S. 9)

- [Pas11] PASSOW, Peter: *Implementierung eines Simulators für die effiziente Exploration von Hardware-Software-Anwendungen in einem Network-on-Chip*. 2011. – Universität Rostock, Masterarbeit (zitiert auf S. 92 und 98)
- [PD10] PASRICHA, Sudeep ; DUTT, Nikil: *On-Chip Communication Architectures: System on Chip Interconnect*. Elsevier Science, 2010 (Systems on Silicon) (zitiert auf S. 40)
- [Pfe09] PFEIFFER, Stephan: *Implementierung eines Network-on-Chip Simulators unter Berücksichtigung von Zuverlässigkeitsaspekten*. 2009. – Universität Rostock, Projektarbeit (zitiert auf S. 91 und 95)
- [PJ06] PENOLAZZI, Sandro ; JANTSCH, Axel: A High Level Power Model for the Nostrum NoC. In: *Proceedings of the 9th EUROMICRO Conference on Digital System Design*. Washington, DC, USA : IEEE Computer Society, 2006 (DSD), S. 673–676 (zitiert auf S. 93)
- [Pop09] POP, Dragos-Paul: A look at Intel's new Nehalem architecture: The Bloomfield and Lynnfield families and the new Turbo Boost technology. In: *Journal of Information Systems & Operations Management* 3 (2009), Nr. 2, S. 410–416 (zitiert auf S. 15)
- [PZGG06] PANDE, Partha P. ; ZHU, Haibo ; GANGULY, Amlan ; GRECU, Cristian: Energy reduction through crosstalk avoidance coding in NoC paradigm. In: *9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools* IEEE, 2006 (DSD), S. 689–695 (zitiert auf S. 58)
- [RMS03] RAN, Yajun ; MAREK-SADOWSKA, Malgorzata: Crosstalk noise in FPGAs. In: *Proceedings of the 40th Design Automation Conference*. New York, NY, USA : ACM, 2003 (DAC), S. 944–949 (zitiert auf S. 44)
- [Röm53] RÖMPP, H.: *Chemie Lexikon*. Franckh, 1953 (Chemie Lexikon Bd. 2) (zitiert auf S. 19)
- [Sal09] SALMINEN, Erno: *On Design and Comparison of On-Chip Networks*, Tampere University of Technology, Diss., 2009 (zitiert auf S. 30)
- [SC02] SOTIRIADIS, Paul-Peter ; CHANDRAKASAN, Anantha P.: A bus energy model for deep submicron technology. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 10 (2002), Nr. 3, S. 341–350 (zitiert auf S. 42)
- [SCH<sup>+</sup>00] SU, Chauchin ; CHEN, Yue tsang ; HUANG, Mu-Jeng ; CHEN, Gen nan ; LEE, Chung len: All digital built-in delay and crosstalk measurement for on-chip buses. In: *Proceedings of the Conference on Design, Automation and Test in Europe, 2000 (DATE)*, S. 527–533 (zitiert auf S. 44)
- [SCK<sup>+</sup>12] SUN, Chen ; CHEN, Chia-Hsin O. ; KURIAN, George ; WEI, Lan ; MILLER, Jason ; AGARWAL, Anant ; PEH, Li-Shiuan ; STOJANOVIC, Vladimir: DSENT - A tool connecting emerging photonics with electronics for opto-electronic Networks-on-Chip modeling. In: *Proceedings of the IEEE/ACM Sixth International Symposium on Networks-on-Chip, 2012 (NOCS)*, S. 201–210 (zitiert auf S. 40 und 84)

- [SCS<sup>+</sup>08] SEILER, Larry ; CARMEAN, Doug ; SPRANGLE, Eric ; FORSYTH, Tom ; ABRASH, Michael ; DUBEY, Pradeep ; JUNKINS, Stephen ; LAKE, Adam ; SUGERMAN, Jeremy ; CAVIN, Robert ; ESPASA, Roger ; GROCHOWSKI, Ed ; JUAN, Toni ; HANRAHAN, Pat: Larrabee: a many-core x86 architecture for visual computing. In: *ACM SIGGRAPH*. New York, NY, USA : ACM, 2008, S. 18:1–18:15 (zitiert auf S. 35)
- [Shu12] SHULER, Kurt: *The Gartner Hype Cycle & Technology Adoption Lifecycle Explained (using NoC Technology)*. Arteris, 2012 (zitiert auf S. 32)
- [SJJ<sup>+</sup>11] SALIHUNDAM, Praveen ; JAIN, Shailendra ; JACOB, Tiju ; KUMAR, Shasi ; ERRAGUNTALA, Vasantha ; HOSKOTE, Yatin ; VANGAL, Sriram ; RUHL, Gregory ; MEMBER ; BORKAR, Nitin: A 2 Tb/s 6 x 4 mesh network for a single-chip cloud computer with DVFS in 45 nm CMOS. In: *IEEE Journal of Solid-State Circuits* 46 (2011), Nr. 4, S. 757–766 (zitiert auf S. 34)
- [SKH08] SALMINEN, Erno ; KULMALA, Ari ; HÄMÄLÄINEN, Timo D.: Survey of Network-on-chip Proposals. In: *White Paper, OCP-IP* (2008) (zitiert auf S. 33, 62, 63 und 64)
- [SLSN98] SINGH, Adit D. ; LAKIN, David R. ; SINHA, Gaurav ; NIGH, Phi: Binning for IC quality: experimental studies on the SEMATECH data. In: *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 1998, S. 4–10 (zitiert auf S. 8)
- [SML98] STAMPER, A.K. ; MCDEVITT, T.L. ; LUCE, S. L.: Sub-0.25-micron interconnection scaling: damascene copper versus subtractive aluminum. In: *Advanced Semiconductor Manufacturing Conference and Workshop, IEEE/SEMI*, 1998, S. 337–346 (zitiert auf S. 38)
- [SS05] SRIDHARA, Srinivasa R. ; SHANBHAG, Naresh R.: Coding for system-on-chip networks: a unified framework. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13 (2005), Nr. 6, S. 655–667 (zitiert auf S. 42)
- [SSH<sup>+</sup>03] SKADRON, Kevin ; STAN, Mircea R. ; HUANG, Wei ; VELUSAMY, Sivakumar ; SANKARANARAYANAN, Karthik ; TARJAN, David: Temperature-aware microarchitecture. In: *ACM SIGARCH Computer Architecture News* Bd. 31 ACM, 2003, S. 2–13 (zitiert auf S. 105)
- [SWY13] SHI, Feng ; WU, Xuebin ; YAN, Zhiyuan: Improved analytical delay models for RC-coupled interconnects. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2013) (zitiert auf S. 42)
- [TB12] TRAN, Anh T. ; BAAS, Bevan: NoCTweak: a Highly Parameterizable Simulator for Early Exploration of Performance and Energy of Networks On-Chip / VLSI Computation Lab, ECE Department, University of California, Davis. 2012 (ECE-VCL-2012-2). – Forschungsbericht. – <http://www.ece.ucdavis.edu/vcl/pubs/2012.07.techreport.noctweak/> (zitiert auf S. 93)

- [TCBS09] TUDORAN, Radu ; CRET, Octavian ; BĂNESCU, Sebastian ; SUCIU, Alin: Implementing true random number generators by generating crosstalk effects in FPGA chips. In: *Proceedings of the 6th FPGAworld Conference*. New York, NY, USA : ACM, 2009 (FPGAworld), S. 25–31 (zitiert auf S. 45)
- [TCST10] TOCKHORN, Andreas ; CORNELIUS, Claas ; SÄMROW, Hagen ; TIMMERMAN, Dirk: Modeling temperature distribution in Networks-on-Chip using RC-circuits. In: *13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2010 (DDECS) (zitiert auf S. 103)
- [TG12] TOCKHORN, Andreas ; GORSKI, Philipp: *Persönliche Mitteilung*. 9 2012 (zitiert auf S. 97 und 98)
- [Til13] TILERA: <http://www.tilera.com/>. Abruf am 13.11.2013 (zitiert auf S. 33 und 65)
- [TKM<sup>+</sup>02] TAYLOR, Michael B. ; KIM, Jason ; MILLER, Jason ; WENTZLAFF, David ; GHODRAT, Fae ; GREENWALD, Ben ; HOFFMAN, Henry ; JOHNSON, Paul ; LEE, Jae-Wook ; LEE, Walter ; MA, Albert ; SARAF, Arvind ; SENESKI, Mark ; SHNIDMAN, Nathan ; STRUMPEN, Volker ; FRANK, Matt ; AMARASINGHE, Saman ; AGARWAL, Anant: The Raw microprocessor: A computational fabric for software circuits and general-purpose programs. In: *IEEE Micro* 22 (2002), Nr. 2, S. 25–35 (zitiert auf S. 33)
- [TTIM08] TANAKA, Genichi ; TAKEUCHI, Kan ; ITO, Minoru ; MATSUSHITA, Hiroaki: A voltage drop aware crosstalk measurement with multi-aggressors in 65nm process. In: *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2008 (CICC), S. 37–40 (zitiert auf S. 44)
- [VHR<sup>+</sup>08] VANGAL, Sriram R. ; HOWARD, Jason ; RUHL, Gregory ; DIGHE, Saurabh ; WILSON, Howard ; TSCHANZ, James ; FINAN, David ; SINGH, Arvind ; JACOB, Tiju ; JAIN, Shailendra ; ERRAGUNTLA, Vasantha ; ROBERTS, Clark ; HOSKOTE, Yatin ; BORKAR, Nitin ; BORKAR, Shekhar: An 80-tile sub-100-W TeraFLOPS processor in 65-nm CMOS. In: *IEEE Journal of Solid-State Circuits* 43 (2008), Nr. 1, S. 29–41 (zitiert auf S. 34)
- [Was12] WASCHKI, Ansgar: *Entwicklung und Integration der Energieprofile von Leitungscodes in die High Level Systemsimulation eines Network-on-Chip*. 2012. – Universität Rostock, Masterarbeit (zitiert auf S. 58)
- [WCG<sup>+</sup>10] WEGNER, Tim ; CORNELIUS, Claas ; GAG, Martin ; TOCKHORN, Andreas ; UHRMACH, Adelinde: Simulation of Thermal Behavior for Networks-on-Chip. In: *28th NORCHIP Conference*, 2010, S. 1–4 (zitiert auf S. 92 und 104)
- [WCTT11] WEGNER, Tim ; CORNELIUS, Claas ; TOCKHORN, Andreas ; TIMMERMAN, Dirk: Monitoring and control of temperature in Networks-on-Chip. In: *Sixth Doctoral Workshop on Mathematical and Engineering Methods in Computer Science – Selected Papers, OpenAccess Series in Informatics*, 2011 (MEMICS'10 & OASIs'11), S. 124–131 (zitiert auf S. 115)

- [WGT11] WEGNER, Tim ; GAG, Martin ; TIMMERMANN, Dirk: Impact of proactive temperature management on performance of Networks-on-Chip. In: *International Symposium on System on Chip*, 2011 (SoC), S. 116–121 (zitiert auf S. 92 und 101)
- [WGT12] WEGNER, Tim ; GAG, Martin ; TIMMERMANN, Dirk: Performance Analysis of Temperature Management Approaches in Networks-on-Chip. In: *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)* 3 (2012), 12, Nr. 4, S. 19–41 (zitiert auf S. 101)
- [WGTU11] WEGNER, Tim ; GAG, Martin ; TIMMERMANN, Dirk ; UHRMACHER, Adelinde: Reduction of thermal imbalances and hot spots in Networks-on-Chip using proactive temperature management. In: *GMM/GI/ITG- Fachtagung Zuverlässigkeit und Entwurf*, 2011 (ZuE), S. 84–91 (zitiert auf S. 92 und 115)
- [WH10] WESTE, Neil ; HARRIS, David: *CMOS VLSI Design: A Circuits and Systems Perspective*. USA : Addison-Wesley Publishing Company, 2010 (zitiert auf S. 11 und 58)
- [Wil01] WILTON, Steven J. E.: A crosstalk-aware timing-driven router for FPGAs. In: *Proceedings of the ACM/SIGDA Ninth International Symposium on Field Programmable Gate Arrays*. New York, NY, USA : ACM, 2001 (FPGA), S. 21–28 (zitiert auf S. 44 und 59)
- [WJ96] WILTON, Steven J. E. ; JOUPPI, Norman P.: CACTI: an enhanced cache access and cycle time model. In: *IEEE Journal of Solid-State Circuits* 31 (1996), May, Nr. 5, S. 677–688 (zitiert auf S. 84)
- [WLM00] WONG, Shyh-Chyi ; LEE, Gwo-Yann ; MA, Dye-Jyun: Modeling of interconnect capacitance, delay, and crosstalk in VLSI. In: *IEEE Transactions on Semiconductor Manufacturing* 13 (2000), Nr. 1, S. 108–111 (zitiert auf S. 39 und 52)
- [WZPM02] WANG, Hang-Sheng ; ZHU, Xinping ; PEH, Li-Shiuan ; MALIK, Sharad: Orion: A power-performance simulator for interconnection networks. In: *Proceedings of the 35th ACM/IEEE International Symposium on Microarchitecture*, 2002 (MICRO), S. 294–305 (zitiert auf S. 84)
- [Xil08] XILINX: *Whitepaper 323, Signal Integrity: Tips and Tricks*. 2008 (zitiert auf S. 45)
- [YCPC10] YOON, Young J. ; CONGER, Nicola ; PETRACCA, Michele ; CARLONI, Luca: Virtual channels vs. multiple physical networks: A comparative analysis. In: *Proceedings of the 47th Design Automation Conference*. New York, NY, USA : ACM, 2010 (DAC), S. 162–165 (zitiert auf S. 65)
- [ZC07] ZHAO, Wei ; CAO, Yu: Predictive technology model for nano-CMOS design exploration. In: *Journal on Emerging Technologies in Computing Systems* 3 (2007), Nr. 1 (zitiert auf S. 52)
- [ZH06] ZHANG, Young ; HUANG, Peicheng ; ZHU, Renjie: Upgrading of integration of time to digit converter on a single FPGA. In: *15th International Workshop on Laser Ranging*, 2006, S. 1–7 (zitiert auf S. 45)

- [ZMLBYT09] ZHANG-MING, Zhu ; LI-BO, Qian ; YIN-TANG, Yang: A novel interconnect-optimal repeater insertion model with target delay constraint in 65 nm CMOS. In: *Chinese Physics B* 18 (2009), Nr. 3, S. 1188–1193 (zitiert auf S. 40)
- [ZWZG12] ZHOU, Fang ; WU, Ning ; ZHANG, Ying ; GE, Xin: NoC router power macro-modeling at high level. In: ZENG, Dehuai (Hrsg.): *Advances in Computer Science and Engineering* Bd. 141. 2012, S. 199–206 (zitiert auf S. 122)

